

Universidad de las Ciencias Informáticas



Facultad 4

Propuesta de Procedimiento para evaluar
atributos internos de software aplicado a la
Herramienta de Autor Web CRODA

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

Autor(a):

Arianna Celia Frenes Padilla

Tutoras:

Ing. Lilian Vigoa Machin

Ing. Yuniet del Carmen Toll Palma

La Habana, junio, 2011

“Año 53 de la Revolución”

Declaración de autoría

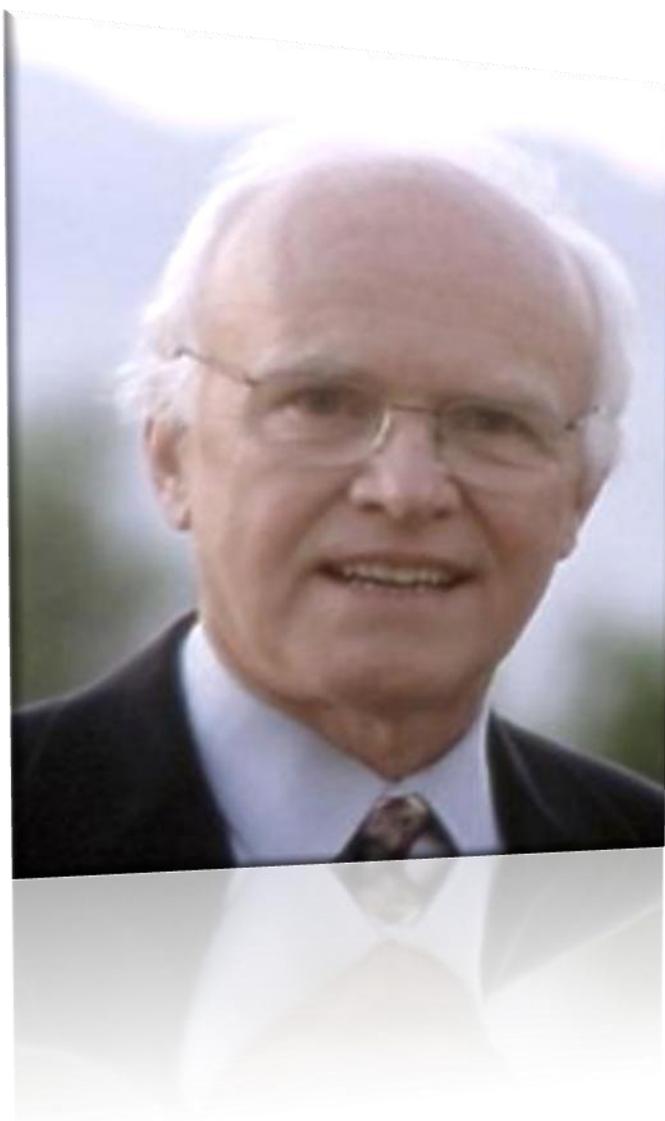
Declaro ser autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ de 2011.

Firma de la Autora

Firma de la Tutora _____

Firma de la Tutora _____



“No podemos controlar lo que no podemos medir”.

Tom De Marco

Dedicatoria

Dedico esta tesis a mi mami y a mi papi que con amor y sabiduría me han mostrado el camino a seguir.

Agradecimientos

A mi madre, por ser la luz de mi vida, la mujer más fuerte que conozco, por su amor y sus sacrificios. No necesitas de títulos ni altas costumbres, te amo como eres porque eres la razón de mí ser. A mi papi, por ser el mejor del mundo. Aunque no te lo diga a menudo, te quiero con la vida y siempre pienso en esas tantas gotas de sudor que derramas por mí y por mis hermanas. Gracias por el impulso y apoyo que me has dado para convertirme en Ingeniera.

A mis hermanas Carmencita y Clari que aunque no se criaron ni han vivido conmigo, se de todo su esfuerzo por hacerme sentir feliz y apoyarme en todo, las quiero mucho. A mi hermano Ariel, porque aunque ha estado ausente la mayor parte de mi vida, me ha enseñado el valor de no separar los sentimientos, pese a la distancia. Espero que algún día te vuelva a ver.

A mis mejores amigos, especialmente a Mayito, Roly, Hismel, Robe, Anita, Yuri, Naglys, Ayled, Roger, Mori y la Mirian. Los quiero mucho y aprecio día tras día todo el cariño y apoyo que me dan. Gracias por soportar mis malcriadeces. A Dany por su cariño, y por soportar mis pesadeces.

A toda mi familia, especialmente a tía Eloina, a Mima, a tío Pio, tío Felo y tío Senaido, por siempre tenerme presentes en su mente y corazón aunque no pueda verlos a menudo.

A las chicas que han convivido conmigo desde mi primer año, a mis queridos compañeros de los distintos grupos en que he estado. A mis profesores especialmente a Desagües y a Mayi, más que profesores son grandes amigos y han estado a mi lado en momentos bien difíciles, ayudándome a no perder la luz del camino. A todos los que han compartido de una forma u otra conmigo, tanto mis tristezas como mis alegrías. A mis tutoras, a la oponente y a los miembros del tribunal por ser justos y ayudarme con cada recomendación hecha. A todos los que contribuyeron de una forma u otra con el desarrollo de este Trabajo de Diploma, especialmente a Dunia, por su disponibilidad. A todos los que preguntaron algún día ¿Cómo va tu tesis?, por su preocupación muchas gracias.

Resumen

Actualmente resulta fundamental para cada empresa productora de software que sus productos cuenten con un elevado prestigio, esto le puede asegurar un lugar reconocido en la industria a nivel mundial. En este contexto se hace necesaria la explotación al máximo de todos los medios que ayuden a garantizar el incremento de la calidad de los productos de software desarrollados. El objetivo fundamental de esta investigación es elaborar un Procedimiento para evaluar atributos internos de software, su aplicación en CRODA permitirá satisfacer las necesidades informativas existentes, para contar con la fundamentación necesaria que pueda guiar al jefe del proyecto a mejorar la toma de decisiones con respecto al manejo de los atributos. Esto permitirá incrementar el nivel de calidad del producto final. El estudio de los atributos de software ha dado paso a identificar los más necesarios para el proyecto CRODA, por otra parte el análisis de las métricas de software ha posibilitado seleccionar las correspondientes para obtener un valor cuantitativo para cada atributo identificado. Los valores alcanzados dan paso a la evaluación, mediante el método estudio de caso, no solo de los atributos sino del proyecto en general. En esta investigación el procedimiento se enfoca en CRODA pero puede adaptarse a otros proyectos que presenten la misma necesidad informativa y no se encuentren incluidos en el Programa de Mejora.

Palabras clave

Calidad de software, Procedimiento, atributos, métricas.

INTRODUCCIÓN.....	7
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	7
1.1 INTRODUCCIÓN	7
1.2 CALIDAD DE SOFTWARE	7
1.3 PROCESO DE MEJORA	10
1.4 ATRIBUTOS DE SOFTWARE	12
1.5 MÉTRICAS DE SOFTWARE.....	15
1.5.1 Métricas de software. Clasificación.....	17
1.5.2 Métricas de software aplicadas a CRODA.....	21
1.6 PROCEDIMIENTO	27
1.7 CONCLUSIONES DEL CAPÍTULO	29
CAPÍTULO II. PROPUESTA DE PROCEDIMIENTO PARA EVALUAR ATRIBUTOS INTERNOS DE SOFTWARE.....	31
2.1 INTRODUCCIÓN	31
2.2 PROCEDIMIENTO. DEFINICIÓN	31
2.3 PROCEDIMIENTO. ESTRUCTURA.....	31
2. 4 PROCEDIMIENTO. ENTRADAS Y SALIDAS.....	34
2. 5 CONCLUSIONES DEL CAPÍTULO	35
CAPÍTULO III. VALIDACIÓN DEL PROCEDIMIENTO PARA EVALUAR ATRIBUTOS INTERNOS DE SOFTWARE APLICADO A CRODA.....	37
3.1 INTRODUCCIÓN	37
3.2 RESULTADOS DE LAS MÉTRICAS CORRESPONDIENTES A LOS ATRIBUTOS.....	37
3.3 MÉTODO DE VALIDACIÓN DEL PROCEDIMIENTO	39
3.3.1 Estudio de Caso. Diseño	41
3.4 ANÁLISIS DE LA EVALUACIÓN DE LOS ATRIBUTOS	44
3.5 RESULTADOS DE LA EVALUACIÓN DE CRODA A PARTIR DEL PROCEDIMIENTO PROPUESTO	49
3.6 CONCLUSIONES DEL CAPÍTULO	57
CONCLUSIONES GENERALES.....	58
RECOMENDACIONES	60
REFERENCIAS BIBLIOGRÁFICAS.....	61

Índice

BIBLIOGRAFÍA.....	63
GLOSARIO DE TÉRMINOS.....	67
ANEXOS.....	70
ANEXO 1. ENCUESTA SOBRE CASOS DE USO DEL PROYECTO	70

INTRODUCCIÓN

Con la aparición y el desarrollo constante de las Tecnologías para la Información y las Comunicaciones (TICs), la humanidad ha dado un gran salto. Hoy en día existe un término que guarda una estrecha relación con lo mencionado anteriormente: el software.

Desde su surgimiento hasta el momento existe un significativo incremento en la utilización y producción de software, convirtiéndose en una importante industria a nivel mundial. Esta situación provoca que la competencia sea día a día mayor en este campo, por tanto es necesario acudir a todas las vías existentes para asegurar el más alto grado de calidad posible y así obtener mejores productos y alcanzar el nivel requerido para la competencia. Se conoce que un error en un software no solucionado a tiempo puede representar la pérdida total del prestigio alcanzado o la imposibilidad o dificultad en el mejor de los casos de alcanzar el mismo (Vega Lebrun y otros, 2008).

El Instituto de Ingenieros Eléctricos y de Electrónicos, o como es conocido por sus siglas en inglés IEEE (Institute of Electrical and Electronics Engineers) define a la calidad del software como: “Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas” (Fuentes Castro, 2008). Por su parte para la organización internacional de estándares ISO/IEC DEC 9126 la calidad de software consiste en “la totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas” (Vega Lebrun y otros, 2008).

Conociendo estos conceptos se puede definir a la calidad del software como un parámetro que ayuda a entender en qué medida, este cuenta con lo requerido. Las métricas de software constituyen herramientas que permiten la obtención de la información necesaria que conllevará a mejorar la toma de decisiones en cuanto al manejo del proyecto, para así lograr obtener un elevado nivel de calidad del software.

En la actualidad el uso de las métricas de software se está poniendo en práctica con éxito en el mercado del software, pues las empresas productoras han reconocido la importancia que tienen para cuantificar los valores de algunos elementos como los atributos de software, lo que permite gestionar de forma más efectiva la calidad de los procesos y

Introducción

productos de software. Una métrica de software se define como un instrumento que permite obtener los valores cuantitativos de los atributos (Serrano y otros, 2005).

Con la introducción de las nuevas tecnologías en los últimos años, la industria del software cubano ha crecido y se ha estimulado su uso en campos como la medicina, educación, industria y otros ámbitos. Esto ha sido posible, sobre todo, a partir de la creación de la Universidad de las Ciencias Informáticas (UCI) a inicios de la pasada década. Este proyecto tuvo como objetivo expreso, potenciar en Cuba la ciencia y la aplicación de la informática en distintas esferas, tanto a nivel nacional como internacional, al aportar al país un nuevo rubro de exportación de productos y servicios, con alto valor agregado.

En 2007 surge, dentro de los perfiles de desarrollo de la universidad, el de Software Educativo. El progreso de este polo, tuvo un punto de inflexión en el año 2010, cuando se convierte en el Centro de Tecnologías para la Formación (FORTES). Esta reestructuración se debe a la búsqueda de una mayor coherencia en las líneas temáticas de desarrollo, con el objetivo de optimizar los esfuerzos productivos y los resultados exportadores de la UCI.

FORTES desarrolla tecnologías que “permiten ofrecer servicios y productos para la implementación de soluciones de formación (...) a todo tipo de instituciones, con diferentes modelos de formación y condiciones tecnológicas (...) a partir de investigaciones que combinan los elementos pedagógicos y tecnológicos más avanzados” (Centro de Tecnologías para la Formación, 2010).

El centro se encuentra compuesto por tres departamentos que se enuncian a continuación:

1. Implantación y Soporte.
2. Producción de Materiales Educativos.
3. Producción de Herramientas Educativas.

Introducción

En este último se encuentra el proyecto CRODA (Creando Objetos de Aprendizaje), en el cual se desarrolla la Herramienta de Autor del mismo nombre. Esta herramienta permite la creación de Objetos de Aprendizaje (OA) reutilizables, accesibles, duraderos e interoperables, de forma flexible, empleando los estándares SCORM (Sharable Content Object Reference Model) y LOM (Learning Object Metadata). Entre los elementos que componen un OA creado a partir de CRODA se encuentran: imágenes, audios, vídeos, links, ecuaciones matemáticas y variados ejercicios de autoevaluación, entre otros (Universidad de las Ciencias Informáticas, 2010).

Existen actualmente dos versiones de CRODA. La versión 1.0 se compone por los siguientes módulos: Usuario, Contenido, Administración, Mensajería y Plantilla; la misma fue registrada por el Centro Nacional de Derecho de Autor y además fue instalada en el Ministerio del Poder Popular para la Educación Universitaria (MPPES) de Venezuela. La versión 2.0, contiene 4 módulos completamente nuevos que son: Diseño Instruccional, Metadatos, Entorno 2.0 y SCORM 2004. El módulo Contenido fue renombrado y en este momento se conoce como Actividades. A pesar de que aún esta versión no ha sido registrada, consta de diversos reconocimientos en eventos como: Jornada Científica Estudiantil, tanto a nivel de facultad como a nivel nacional, Fórum de Ciencia y Técnica, a nivel de facultad y Universidad 2012, a nivel de facultad.

En la actualidad existen muchas herramientas para garantizar que el software alcance un alto nivel de calidad como por ejemplo, métricas, normas, procedimientos, etc. Sin embargo, a partir de las entrevistas con la dirección y los miembros del proyecto CRODA, fue posible diagnosticar que estas herramientas no son utilizadas en todo su potencial, lo cual afecta los resultados de calidad que se esperan del proyecto.

Es necesario para CRODA 2.0 conocer las evaluaciones de un conjunto de atributos internos de software que han sido identificados como principales por el jefe de proyecto, los cuales actualmente se refieren a: *personal, duración, costo, tamaño, defectos, errores, madurez, esfuerzo y productividad*. La importancia de estos radica, no solo en la necesidad informativa que cubren, sino también en que facilitan el trabajo de los dos revisores técnicos del proyecto, los cuales están encargados de realizar las mediciones de los atributos, debido a que la evaluación de estos atributos no es altamente compleja.

Introducción

A raíz del Programa de Mejora que se lleva a cabo en la UCI desde finales de 2008, algunos artefactos del proyecto fueron modificados, como por ejemplo el Plan de Mediciones, el cual fue eliminado. Tomando en consideración lo expuesto junto a la necesidad para CRODA de irse adaptando al Programa de Mejora, se considera que no debe contarse con un Plan de Mediciones para la versión actual.

Sin embargo, sigue siendo importante para el control interno del proyecto, contar con información suficiente y actualizada, que muestre el estado en que se encuentran los atributos identificados, con el objetivo fundamental de garantizar su mejora continua. En CRODA 2.0 la información que permite respaldar la toma de decisiones no es suficiente, esto trae consigo la imposibilidad de afirmar con certeza que las decisiones tomadas para incrementar la calidad del producto son correctas, además aumenta potencialmente el riesgo de fracaso del proyecto e incluso podría dar paso a la obtención de un producto que no sea precisamente el que se espera y a la aparición de pérdidas significativas de recursos.

Teniendo en cuenta la situación planteada anteriormente surge el siguiente **problema de investigación**: ¿Cómo evaluar los atributos internos identificados como principales del proyecto CRODA? Teniendo como **objeto de estudio**: las métricas de software y como **campo de acción**: las métricas de software aplicadas en el proyecto CRODA.

Se define como **objetivo general** de este trabajo de diploma: elaborar un Procedimiento que indique cómo evaluar los atributos internos de software identificados como principales por la dirección del proyecto CRODA.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

- ❖ Elaborar el marco teórico de la investigación.
- ❖ Describir la propuesta del Procedimiento para evaluar los atributos internos de software identificados como principales por el jefe del proyecto CRODA.
- ❖ Validar el procedimiento propuesto, a partir de un estudio de caso múltiple, aplicado al proyecto CRODA

Introducción

En la presente investigación se define la siguiente **idea a defender**: la elaboración de un Procedimiento para evaluar los atributos internos de software identificados como principales por el jefe del proyecto CRODA, mostrará el estado de los atributos evaluados y apoyará la toma de decisiones para incrementar la calidad en el mismo.

Para darle cumplimiento a los objetivos específicos enunciados anteriormente se han establecido como **tareas de investigación**:

- ❖ Estudio de las clasificaciones de métricas para medir atributos de software a nivel mundial.
- ❖ Análisis de las métricas para medir los atributos de software en la UCI.
- ❖ Selección de las métricas de software a utilizar en el proyecto CRODA.
- ❖ Diseño y elaboración de un Procedimiento para evaluar atributos internos de software.
- ❖ Evaluación de los atributos del proyecto CRODA.
- ❖ Validación del Procedimiento para evaluar atributos internos de software del análisis de los resultados de la medición de los principales atributos en el proyecto CRODA.

Para la realización del presente trabajo de diploma se utilizaron los **Métodos Científicos de Investigación** que se enuncian a continuación:

Los **métodos teóricos** permiten comprender la situación que se estudia, su evolución y proponer las mejoras a los problemas que se identificaron, dentro de estos métodos se encuentran los siguientes, que fueron utilizados durante la investigación:

Analítico – sintético: Al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.

Análisis Histórico – lógico: Para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.

Introducción

Método Explicación: Para la elaboración de un procedimiento en el cual se requiere de una explicación del porqué y el cómo se hará cada paso del mismo.

Los **métodos empíricos** permiten describir y explicar las características del fenómeno en estudio, específicamente como métodos de esta clase fueron utilizados:

Observación: Al registrar visualmente lo que ocurre en realidad con respecto al tema directamente sin intermediarios. Se utilizó la observación participante en el diagnóstico de la situación problemática y la validación de la propuesta.

Entrevistas: Al establecer los elementos necesarios, tanto para conocer sobre los temas relacionados con las métricas de software, como para encontrar argumentos en el momento de diseñar las mismas.

El presente documento se encuentra estructurado de la siguiente forma:

Capítulo 1: Se presentan los fundamentos teóricos que sustentan la investigación. Se enfoca de forma general en analizar la teoría referente al tema de la calidad de software, específicamente sobre métricas; así como los conceptos que estén abiertamente relacionados con el tema, siendo este capítulo la base teórica para la comprensión de la investigación.

Capítulo 2: A partir de los estudios del capítulo anterior se desarrolla la propuesta de Procedimiento para evaluar los atributos internos de software del proyecto CRODA; definiendo las fases y procesos que estarán presentes en el mismo.

Capítulo 3: En este capítulo se muestra la validación de los resultados obtenidos de la propuesta, empleando el método de evaluación cualitativa estudio de caso en su variante múltiple, mediante su aplicación al proyecto CRODA.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

Actualmente para Cuba, ganar reconocimiento en la industria y la ciencia del software, a nivel mundial, constituye una necesidad. En este capítulo se recoge la más importante teoría referente al tema de la calidad de software, específicamente sobre métricas, las cuales constituyen un importante factor para asegurar la calidad de los productos realizados.

1.2 CALIDAD DE SOFTWARE

Existen diversos conceptos de calidad, que varían en función del área del conocimiento donde se aplican y en las épocas en las que han sido definidos. Se puede observar la evolución de la definición de calidad desde que se denominó como: “aquella que el productor es capaz de darle al cliente en conformidad con las especificaciones de su producto” hasta “aquella que se adecua a las necesidades de los consumidores, y se asocia con el uso y valor que da satisfacción a dichas necesidades” (Romero y otros, 2007).

La norma ISO 9000:1994, define la calidad como la “totalidad de (...) características de una entidad que influyen en su capacidad para satisfacer necesidades establecidas e implícitas”. En la última versión de la ISO 9000 correspondiente al año 2000, queda como el “grado en el que el conjunto de características inherentes cumple con los requisitos” (Romero y otros, 2007).

Constituye un criterio común, situar la calidad como encaminada a lograr la satisfacción plena y total del cliente. A nivel mundial, alcanzar un máximo índice en lo que a este respecta se ha convertido en una máxima prioridad. Aunque se conozcan diversos conceptos de calidad siempre resulta provechoso buscar el que más se adapte a cada situación.

De manera general, en la bibliografía estudiada pudo observarse que cada concepto de calidad se enfoca según los intereses, experiencias y subjetividades de su creador. A

Capítulo I. Fundamentación Teórica

pesar que el planteamiento de la norma ISO 9000:1994 no se considera incorrecto, se entiende como más acertado el ofrecido por la norma ISO 9000 en el año 2000. Bajo esta última referencia, resulta ser el nivel en el que los atributos primordiales de un elemento cumplen con lo que se requiere, haciendo así que este en su totalidad cumpla con lo que se necesita o espera. Teniendo en cuenta lo anterior, es posible inferir que este concepto se ajusta más al tema de la calidad cuando se refiere específicamente a la industria del software.

La calidad de un producto de software se ha definido como la “concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente que desea el usuario” Roger S. Pressman (1998). El Instituto de Ingenieros Eléctricos y Electrónicos en el año 1990 (IEEE Std. 610-1990), también brindó su aporte al enunciar que: “La calidad del software es el grado con el que un sistema, componente o proceso cumple con requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Luego de analizar los conceptos expuestos sobre el tema de calidad de software se puede afirmar que el más completo es el enunciado por Roger S. Pressman. Además de que resulta más actual, este se enfoca no solo en la importancia que le adjudica a las especificaciones del cliente, sino, también en el cumplimiento de lo establecido por los estándares de calidad existentes. Partiendo de este análisis se puede definir a la calidad del software como el nivel en que una aplicación informática cumple satisfactoriamente con las expectativas del usuario y con los estándares establecidos.

Se considera que la calidad presente en un software debe contar con un valor real que represente el estado de los atributos del proyecto, este puede calificarse de Bien, Regular o Mal. Con el conocimiento del estado de los principales atributos del proyecto se puede establecer el estado del proyecto en general. Esta información obtenida permitirá conocer el estado de cada atributo con el objetivo fundamental de que el software logre alcanzar un nivel de calidad óptimo.

Los atributos de calidad de software según lo planteado por Bell en el año 2000 son:

- ❖ Fiable: Capacidad de ofrecer los mismos resultados bajo las mismas condiciones.

Capítulo I. Fundamentación Teórica

- ❖ Eficiente: Utilización óptima de los recursos de la máquina.
- ❖ Robusto: No poseer un comportamiento catastrófico ante situaciones excepcionales (Tolerante a fallos).
- ❖ Correcto: Se ajusta a las especificaciones dadas por el usuario.
- ❖ Portable: Capaz de integrarse en entornos distintos con el mismo esfuerzo.
- ❖ Adaptable (extensibilidad): Modificar alguna función sin que afecte a sus actividades.
- ❖ Inteligible: Diseño claro, bien estructurado y documentado.
- ❖ No erróneo: No exista diferencia entre los valores reales y los calculados.
- ❖ Reutilizable (reusabilidad).

Sommerville por su parte, en el año 2002, define los atributos de calidad del software como:

- ❖ Mantenimiento.
 - Fiabilidad.
 - Seguridad.
 - Protección.
- ❖ Confiabilidad.
 - Fiabilidad.
 - Seguridad.
 - Protección.
- ❖ Eficiencia.
- ❖ Usabilidad.

La información necesaria que debe contribuir al mejoramiento de los atributos, se obtiene mediante la utilización de los modelos de calidad de software definidos. La construcción de un modelo no es nada fácil, usualmente estos dividen la calidad del software en una serie de características que pueden ser útiles al comprobar los aspectos relacionados con la calidad. La mayor parte de los modelos se encuentran basados en la norma ISO9126. Esta norma define un grupo de características de calidad de las cuales se derivan otras,

Capítulo I. Fundamentación Teórica

estas a su vez se descomponen en atributos. Los valores de estos atributos se calculan mediante la utilización de métricas (Calero Muñoz, 2005).

Entre los modelos basados en la norma ISO 9126, se encuentra el modelo propuesto por Bertoa y Vallecillo (2002) para componentes de software. En este los autores adaptan la norma ISO9126 a los componentes COTS (Comercial-Off-The-Shelf). El término COTS se refiere a aquel software comercial desarrollado previamente por terceras partes, fuera de las estrategias de desarrollo. El modelo de calidad QUINT2 (Niessink, 2002) también presenta una ampliación de la norma ISO 9126, pensada para valorar la calidad de arquitecturas de software. Por su parte el modelo de calidad propuesto por Franch y Carvallo (2003) presenta una adaptación de la ISO 9126 para correo electrónico (Calero Muñoz, 2005).

En la UCI actualmente se toma como marco de referencia para mejorar los procesos el Modelo Integrado de Capacidad y Madurez (CMMI). Según lo estudiado hasta el momento, ninguna de las fuentes utilizadas ha arrojado algún resultado concreto que demuestre un posible vínculo entre este modelo y la norma ISO 9126, pero aun así se considera que ambos tienen mucho en común. CMMI no solo descompone la calidad en niveles estructurados e intenta cuantificar la calidad del software, al igual que la norma ISO 9126, sino que también emplea las métricas con el objetivo de medir y analizar los principales atributos del software, lo cual permite adquirir un nivel elevado de conocimiento sobre el estado de los mismos y de esta forma lograr mejorar el estado en el que se encuentra cada uno de ellos. Todo esto posibilita que la calidad del software en general se incremente. CMMI constituye la base del Programa de Mejora que se lleva a cabo en la UCI.

1.3 PROCESO DE MEJORA

La UCI, ha tenido desde su comienzo la meta de alcanzar altos índices de producción e ingresos en la esfera de la informática, por lo que se involucró desde finales del año 2008 en un proceso de mejora de software (SPI por sus siglas en inglés "Software Process Improvement"). El proceso o programa de mejora es un concepto que pretende mejorar los productos, servicios y procesos, el mismo supone importantes beneficios para las organizaciones de software como (Calisoft, 2009):

Capítulo I. Fundamentación Teórica

- ❖ Se produce un incremento de la satisfacción del cliente al utilizar un software, con una cantidad de errores inferior.
- ❖ Se incrementa la eficiencia del proceso de desarrollo.
- ❖ Se facilita la definición y cumplimiento de los objetivos de calidad.
- ❖ Se incrementa la satisfacción de los trabajadores debido a que se proporcionan herramientas y recursos apropiados para la realización eficiente del trabajo.

En una entrevista realizada por Dunieski Sarmiento Méndez a la directora del centro Calisoft, Ailyn Febles, sobre el programa de mejora, esta afirma que CMMI se ha convertido hoy en el modelo líder, en la evaluación de los procesos de desarrollo de software para su mejora. Todo esto permite establecer el grado de importancia que presenta CMMI para las organizaciones productoras de software. El modelo se compone por 5 niveles de madurez y 22 áreas de proceso, el mismo utiliza Standard CMMI Appraisal Method for Process Improvement (SCAMPI) como metodología para evaluar al software según los niveles de madurez.

El programa de mejora está encaminado a que la UCI alcance la certificación internacional del nivel 2 del modelo CMMI, que propone una administración disciplinada de los proyectos, donde se establezcan y sigan políticas organizacionales. Este nivel engloba 7 áreas de proceso (Suárez Batista y otros, 2011):

- ❖ Aseguramiento de la Calidad de Procesos y Productos (PPQA (Process and Product Quality Assurance)).
- ❖ Administración de Requerimientos (REQM).
- ❖ Planeación del Proyecto (PP).
- ❖ Monitoreo y Control de Proyecto (PMC).
- ❖ Medición y Análisis (MA).
- ❖ Administración de Configuración (CM).
- ❖ Administración de Proveedores (SAM).

Capítulo I. Fundamentación Teórica

Cada área de proceso cuenta con un propósito en específico, que se resume en solucionar las deficiencias que pueden existir en un área determinada. Las áreas de proceso o procesos¹ tienen en común el uso de métricas para su desarrollo, esto demuestra la importancia que tienen las mismas para la mejora del software, por lo tanto es necesario priorizar su aplicación. Para CRODA es necesario el uso de métricas para facilitar la obtención de los valores cuantitativos correspondientes a los atributos identificados como principales por el jefe del proyecto.

1.4 ATRIBUTOS DE SOFTWARE

Según la norma ISO 14598-1:1999 un atributo de un proyecto de software no es más que aquella característica medible de este. Los atributos siguen la siguiente clasificación (Departamento de Lenguajes y Sistemas Informáticos, 2009):

- ❖ Internos: Son medidos examinando el proceso, producto o proyecto mismo. Son ejemplos de atributos internos del software: tamaño, errores cometidos, duración, entre otros.
- ❖ Externos: Se miden con respecto a cómo el proceso, producto o proyecto se relaciona con su entorno. Se pueden citar como atributos externos de software: fiabilidad, mantenimiento, usabilidad, entre otros.

En la industria del software existen tres clases de entidades² cuyos atributos puede interesar medir (Calero Muñoz, 2006):

- ❖ Procesos: Son actividades del software que normalmente dependen del factor tiempo. Atributos internos importantes: el tiempo (duración del proceso), el esfuerzo (asociado al proceso) y el número de incidentes de un tipo específico que se dan durante el

¹ Las áreas de proceso del nivel 2 de CMMI están definidas como procesos en el Programa de Mejora.

² Según el estándar internacional ISO 15939, que define un proceso de medición para el desarrollo del software, una entidad no es más que aquel objeto que va a ser caracterizado mediante una medición de sus atributos. Puede ser un proceso, un producto, un servicio, un proyecto, o un recurso concreto.

Capítulo I. Fundamentación Teórica

proceso (por ejemplo el número de errores de requisitos encontrados durante la construcción de la especificación).

- ❖ **Productos:** Entregables, artefactos o documentos generados en el ciclo de vida del software. Ejemplos de atributos externos son: la fiabilidad del código, el mantenimiento del código fuente, entre otros. Como ejemplo de atributos internos pueden ser citados: la longitud, funcionalidad, modularidad o corrección sintáctica de los documentos de especificación, entre otros.
- ❖ **Recursos:** Aquellos elementos que hacen de entrada a la producción del software. Por ejemplo el personal, los materiales, las herramientas y los métodos. Un atributo interesante es el costo. En el caso del personal, además del costo, se suele medir la productividad.

Cada proyecto de software tiene su necesidad informativa particular y a raíz de esto, los jefes de proyecto determinan los atributos cuya evaluación la satisface de la mejor manera posible. Hasta el momento y según lo estudiado en el transcurso de la presente investigación, no existe alguna regla que especifique que todos los proyectos deban tener un mismo grupo de atributos clasificados como principales, por lo tanto se considera que pueden existir coincidencias pero no es obligatorio. En el caso de CRODA han sido identificados por el líder del proyecto como atributos principales:

- ❖ **Personal:** Atributo interno del proyecto cuyo valor cuantitativo está dado por la cantidad de personas que trabajan en este.
- ❖ **Duración:** Atributo interno del proyecto cuyo valor refleja la cantidad de tiempo con la que cuenta este para su desarrollo.
- ❖ **Costo:** Atributo interno del proyecto cuyo valor cuantitativo expresado en unidades monetarias indica cuanto debe pagar el cliente por el producto desarrollado
- ❖ **Tamaño:** Atributo interno del proyecto. De manera general el tamaño de un software se expresa en LOC (cantidad de líneas de código), pero en la UCI se mide según la cantidad de casos de uso por las medidas de tamaño existentes (Tapanes Alfonso, 2008):

Capítulo I. Fundamentación Teórica

- Guiones: Un guión es un caso de uso, en dependencia de la complejidad se clasifica en Alta, Media y Baja.
 - Guión técnico: Explica detalladamente qué contiene cada pantalla.
 - Guión de contenido: Lo hace el cliente o pedagogo, por ejemplo explicando el por qué y para qué se hace la pantalla.
- POP: Medida utilizada en los proyectos de diseño. Son todos los materiales creados en un proyecto de diseño.
- Requisitos: La relación de varios requisitos pueden conformar un caso de uso.
- Historia de usuario: Una historia de usuario es un caso de uso. Para la clasificación del caso de uso hay que extraer del contenido de la descripción, la cantidad de Flujos Básicos y Flujos Alternos e incluirlo en la clasificación de la complejidad de estos según la cantidad. La idea principal es describir un caso de uso en dos o tres líneas con la terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen pruebas de aceptación para el usuario y estas permitan estimar el tiempo de desarrollo del caso de uso.
- BPMN: Se basa en la descripción de procesos. Un proceso puede estar formado por uno o varios casos de uso. Todo depende de la complejidad.
- Módulos: Un módulo está compuesto por varios módulos y a la vez estos están contenidos por uno o varios casos de uso relacionados que se clasificarán en dependencia de la complejidad.

Teniendo en cuenta este análisis se determina que todas estas medidas pueden convertirse en casos de uso como unidad de medida estándar para la UCI. El *tamaño* de un software influye en varios atributos como: *personal, esfuerzo, costo y duración*.

- ❖ Defectos: Atributo interno del proyecto cuyo valor es la cantidad de fallas que continúan presentes en el producto luego de su liberación, las mismas pasan a ser parte del producto incrementando su nivel de deficiencia
- ❖ Errores: Atributo interno del proyecto cuyo valor está dado por la cantidad de fallas encontradas durante las pruebas realizadas al producto o durante el desarrollo del

Capítulo I. Fundamentación Teórica

mismo, esta cifra debe ser disminuida antes de la salida del producto para mejorar la calidad del mismo

- ❖ **Madurez:** Atributo interno del proyecto cuyo valor cuantitativo se interpreta como el grado de estabilidad alcanzado por el software.
- ❖ **Esfuerzo:** Atributo interno del proyecto cuyo valor cuantitativo se interpreta como el trabajo que ha realizado el personal del proyecto en el tiempo de duración del mismo.
- ❖ **Productividad:** Atributo interno del proyecto cuyo valor cuantitativo se interpreta como la relación entre la producción (codificación o documentación) obtenida por un sistema productivo (el proyecto) y los recursos utilizados para obtener dicha producción (personal).

El proyecto CRODA cuenta solamente con dos revisores técnicos (estudiantes) que son los encargados de realizar las mediciones y por ese motivo se seleccionaron solo los atributos que son considerados en su conjunto como simples y pueden proveer una evaluación clara del producto. Por este motivo solamente fueron identificados como principales los expuestos. Para cada uno de los atributos que han sido descritos, será obtenido un valor cuantitativo mediante la utilización de las métricas correspondientes.

1.5 MÉTRICAS DE SOFTWARE

“Una medida facilita una aproximación cuantitativa de la cantidad, dimensiones o tamaño de algunos atributos de un producto. La medición es la acción de establecer una medida” (Calero Muñoz, 2006). Con el apoyo de estas definiciones se puede establecer a una métrica como un instrumento que facilita la obtención de un valor cuantitativo correspondiente a un atributo. Informalmente una métrica también se puede definir como un factor para establecer la diferencia entre dos objetos (La Torre Hernández y otros, 2008).

La importancia del uso de las métricas se ve reflejada en el incremento de su uso para el diagnóstico de la calidad de procesos y productos. En la industria del software el uso de las métricas ha cobrado gran importancia, dada la influencia de estas en el tema de la mejora tanto de procesos como de producto.

Capítulo I. Fundamentación Teórica

La expresión “métrica de software”, puede entenderse como aquellas mediciones basadas en técnicas que se aplican a procesos, productos y servicios brindando una óptima administración de información lo que ayuda a proveer a los procesos, productos y servicios medidos un alto nivel de mejora. Cada métrica de software empleada debe cumplir con determinadas características que respalden su aplicación. Las mismas deben ser sencillas y a la vez robustas, consistentes y fáciles de obtener (La Torre Hernández y otros, 2008).

Actualmente el uso de las métricas de software más que una vía para asegurar la calidad de un producto, es una necesidad, ya que ayudan a realizar estimaciones más precisas, además garantizan una mayor productividad con la obtención de productos finales de mejor calidad. Las métricas de software cuentan con los siguientes principios (Fuentes Castro, 2008):

- ❖ Medir es un mecanismo ideal para caracterizar, evaluar, predecir y proporcionar motivación para los diversos aspectos de los procesos de construcción del software.
- ❖ Las medidas deben aplicarse tanto sobre el proceso software como en el producto.
- ❖ Debe estar claramente indicado el propósito de cada medida.
- ❖ Los entornos de desarrollo y mantenimiento deben estar preparados para las medidas.
- ❖ Se necesitan múltiples mecanismos para la recopilación y validación de datos.
- ❖ Para evaluar y comparar proyectos y para llevar a cabo modelos se necesita una base histórica de experiencias.

Como desventajas pueden mencionarse las siguientes (La Torre Hernández y otros, 2008):

- ❖ Diversidad de métricas que abordan la calidad con criterios diferentes, debido al desacuerdo en los criterios involucrados.

- ❖ Las métricas no proporcionan información *per se* y en lugar de esclarecer, confunden a la contraparte del modelador dentro del proceso.
- ❖ Muchas métricas no guardan relación con los intereses de las partes, y el indicador de la calidad de un esquema, se construye generalmente con todas.

1.5.1 MÉTRICAS DE SOFTWARE. CLASIFICACIÓN

Existen muchas y diversas formas de clasificar las métricas del software, distintas unas de otras según los autores, Roger S. Pressman lo hace en 6 categorías o grupos distintos (Pressman, 2002):

- ❖ Métricas técnicas: Mide la estructura del sistema, el cómo está hecho. Es decir, están centradas en las características del software más que en su proceso de desarrollo.
- ❖ Métricas de calidad: Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente; o sea cómo medir para que el sistema se adapte a los requisitos del cliente.
- ❖ Métricas de productividad: Referidas al rendimiento del proceso de desarrollo como función del esfuerzo aplicado. Se centran en el rendimiento del proceso de la ingeniería del software. En otras palabras, qué tan productivo va a ser el software que se diseñará.
- ❖ Métricas orientadas al tamaño: Muestran en qué periodo de tiempo será terminado el software y cuántas personas se necesitarán. Son medidas directas al software y al proceso que se sigue para su desarrollo.
- ❖ Métricas orientadas a la función: Son medidas indirectas del software y del proceso por el cual se desarrolla. Las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.
- ❖ Métricas orientadas a la persona: Proporcionan medidas e información sobre la forma en la que el personal desarrolla el software.

Capítulo I. Fundamentación Teórica

Otras de las clasificaciones dadas a las métricas de software, se dividen según los criterios y según el contexto donde se aplican. Según los criterios podemos clasificar las métricas de software en (La Torre Hernández y otros, 2008):

- ❖ Métricas de software de complejidad: Definen la medición de la complejidad en cuanto a volumen, tamaño, anidaciones, y configuración.
- ❖ Métricas de software de calidad: Definen la calidad del software como exactitud, estructuración o modularidad, pruebas, mantenimiento.
- ❖ Métricas de software de competencia: Miden las actividades de productividad de los programadores con respecto a su certeza, rapidez, eficiencia y competencia.
- ❖ Métricas de software de desempeño: Miden la conducta de módulos y sistemas de un software, bajo la supervisión del Sistema Operativo o el hardware.
- ❖ Métricas de software estilizadas: Estilo de código, convenciones, limitaciones, etc.

Según el contexto donde se aplican podemos clasificar las métricas de software en:

- ❖ Métricas de proceso: Se recopilan de todos los proyectos, y durante un largo periodo de tiempo. Se caracterizan por el control y ejecución del proyecto y la medición del tiempo de las fases. En las métricas de proceso, el proveedor debe tener métricas cuantitativas de la calidad del proceso de desarrollo y de liberación. Estas métricas deben reflejar:
 - Qué tan bien está siendo llevado a cabo el proceso de desarrollo, teniendo en cuenta los puntos de revisión y el cumplimiento, de los objetivos de calidad en el proceso, en el tiempo adecuado.
 - Qué tan efectivo es el proceso de desarrollo, al reducir la probabilidad que se introduzcan fallas o que cualquier falla introducida sea detectada.

Aquí del mismo modo que las partes de las métricas del producto, lo importante es que los niveles sean conocidos y utilizados para el control del proceso y de las mejoras y no sean utilizadas métricas fijas. Las métricas seleccionadas deben de ajustarse al proceso utilizado y si es posible, tener un impacto directo sobre la calidad de software liberado. Su

Capítulo I. Fundamentación Teórica

intento es proporcionar datos que lleven a mejoras de los procesos de software a largo plazo.

La medición del proceso implica las mediciones de las actividades relacionadas con el software siendo algunos de sus atributos típicos el esfuerzo, el coste y los defectos encontrados. Las métricas permiten tener una visión profunda del proceso de software que ayudará a tomar decisiones mejor fundamentadas, ayudan a analizar el trabajo desarrollado, conocer si se ha mejorado o no con respecto a proyectos anteriores. Ayudan a detectar áreas con problemas para poder remediarlos a tiempo y a realizar mejores estimaciones.

Para mejorar un proceso se deben medir los atributos del mismo, desarrollar métricas de acuerdo a estos atributos y utilizarlas para proporcionar datos concretos que conduzcan la mejora del proceso. Los errores detectados antes de la entrega del software, la productividad, recursos y tiempo consumido y el ajuste con la planificación son algunos de los resultados que pueden medirse en el proceso, así como las tareas específicas de la ingeniería del software.

Las métricas del proceso se caracterizan por el control y ejecución del proyecto, la medición de los tiempos del análisis, diseño, implementación, implantación y pos implantación, la medición de las pruebas (errores, cubrimiento, resultado en número de defectos y número de éxito) y la medición de la transformación o evolución del producto.

- ❖ Métricas de proyecto: Permiten evaluar el estado del proyecto y seguir la pista de los riesgos. Los datos obtenidos según la aplicación de las métricas de proyecto permiten al administrador de software (Pressman, 2002):
 - Evaluar el estado del proyecto en curso.
 - Realizar un seguimiento de los riesgos potenciales.
 - Detectar las áreas de problemas antes de que estos pasen a complicarse al punto de no tener solución.
 - Ajustar el flujo y las tareas de trabajo.

Capítulo I. Fundamentación Teórica

- Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo de la ingeniería del software.

La primera aplicación de métricas de proyectos en la mayoría del software ocurre durante la estimación. Las métricas recopiladas de proyectos anteriores se utilizan como una base desde la que se realizan las estimaciones del esfuerzo y del tiempo para el actual trabajo del software. A medida que avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales.

El gestor de proyectos utiliza estos datos para supervisar y controlar el avance. A medida que comienza el trabajo técnico, otras métricas de proyectos comienzan a tener significado. Se miden los índices de producción representados mediante páginas de documentación, las horas de revisión, los puntos de función y las líneas fuentes entregadas, en el proyecto se sigue la pista de los errores detectados durante todas las tareas de ingeniería del software. Cuando va evolucionando el software desde la especificación del diseño, se recopilan las métricas técnicas para evaluar la calidad del mismo y proporcionar datos que influirán en el enfoque tomado para la generación y prueba del código.

- ❖ Métricas de producto: Se centran en las características del software y no en cómo fue producido. Se miden atributos como el tamaño, la calidad, la totalidad, y el esfuerzo. 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. Un producto no es solo el software o sistema funcionando, sino también los artefactos, documentos, modelos, módulos, o componentes que lo conforman, por tanto, las métricas del producto deben hacerse sobre la base de medir cada uno de estos. En los artefactos del producto se mide, entre otras cosas, el tamaño, la calidad (teniendo en cuenta los defectos, la complejidad, entre otros), la totalidad, rastreabilidad, esfuerzo.

Las métricas, según la bibliografía consultada para realizar esta investigación, pueden ser clasificadas, siguiendo el criterio de composición, en derivadas y bases o primitivas. Las

Capítulo I. Fundamentación Teórica

derivadas son las que utilizan a otras métricas para su cálculo y las bases o primitivas, son las que no dependen de otras para su aplicación.

Con el incremento de la calidad de la toma de decisiones por parte del jefe de proyecto y con respecto a los principales atributos internos; se disminuye significativamente la probabilidad de fracaso del proyecto, aumenta la confiabilidad de las personas que desarrollan el proyecto, hacia su líder, se ahorra tiempo, recursos y energía e indica que los problemas no son pasados por alto; estos son valorados, considerados y solucionados.

Es importante para Cuba alcanzar un alto prestigio por lo que debe mejorar día a día la calidad de los productos realizados, para ello debe emplear hasta el último recurso existente. Aunque aún el tema de las métricas de software no se domina como un término común en la industria cubana se conoce que las mediciones contribuyen al control, seguimiento y mejora de la calidad del proceso de desarrollo de software pero este conocimiento no es suficiente sino cuenta con el respaldo de una aplicación real (La Torre Hernández y otros, 2008).

Existen varias razones para demostrar la importancia del uso de las métricas en el proceso de desarrollo de software. Algunos autores afirman que cuando se puede medir aquello de lo cual se está hablando y se puede expresar en números, se sabe realmente acerca de ello; pero cuando no puede medirse y no puede expresarse en números el conocimiento que se tiene de ello es escaso e insatisfactorio. El contar con datos reales de métricas de software, provee un panorama de situaciones existentes que ayudan a aplicar y dar seguimiento a las diferentes formas de evaluar y determinar métricas de calidad de software para un mejor desempeño.

Tomando como base lo estudiado, se considera que por cada software desarrollado debe existir un grupo de métricas que posibiliten mejorar el estado de los principales atributos del mismo y así de esta forma contribuir a aumentar la calidad del software.

1.5.2 MÉTRICAS DE SOFTWARE APLICADAS A CRODA

En Calisoft se encuentran un grupo de métricas de software clasificadas según el contexto donde se aplican, es decir, en Producto, Proceso y Proyecto y según su composición

Capítulo I. Fundamentación Teórica

(Base y Derivada). Este conjunto de métricas fue analizado y se seleccionaron las correspondientes para obtener el valor cuantitativo de cada atributo identificado como principal por el jefe del proyecto CRODA. Las métricas seleccionadas son:

- ❖ Personal Necesario (PN): La métrica PN está clasificada según su composición como métrica base, se obtiene directamente y el encargado de esto es el revisor técnico mediante la realización de entrevistas al jefe de proyecto. Según el contexto donde se aplica se clasifica esta métrica como de proyecto. Su valor se expresa en personas. En la interpretación de su valor cuantitativo es necesario tener en cuenta otros atributos como *tamaño*.
- ❖ Tiempo de Desarrollo Real (TDESR): La métrica TDESR se encuentra clasificada según su composición como métrica base, se obtiene directamente y el encargado de esto es el revisor técnico mediante la realización de entrevistas al jefe de proyecto. Su valor se expresa en meses y se interpreta como el tiempo en el que ha transcurrido el proyecto desde el inicio hasta la entrega, que en el caso de CRODA se tomará como el momento actual porque aún no se ha liberado el producto. Para interpretar el valor obtenido es necesario tener en cuenta la existencia de atrasos en el cronograma establecido como parte de la negociación.
- ❖ Costo (C): La métrica C está clasificada según su composición como métrica base, se obtiene directamente y el encargado de esto es el revisor técnico mediante la realización de entrevistas al jefe de proyecto. Su valor se expresa en pesos. Según el contexto donde se aplica se clasifica esta métrica como de proyecto. Para su interpretación es necesario tener en cuenta lo negociado con el cliente además de la influencia de varios atributos como *tamaño, personal y duración*.
- ❖ Tamaño (T): La métrica T está clasificada según su composición como métrica base, se obtiene directamente y el encargado de esto es el revisor técnico mediante la realización de entrevistas al jefe de proyecto. El valor se mide en casos de uso especificados. Según el contexto donde se aplica se puede clasificar como métrica de producto. Para su interpretación es necesario tener en cuenta la complejidad de los casos de uso.

Capítulo I. Fundamentación Teórica

- ❖ Defectos (DEFCT): La métrica DEFCT está clasificada según su composición como métrica base, se obtiene directamente y el encargado de esto es el revisor técnico mediante la realización de entrevistas al jefe de proyecto. Su valor se expresa en unidades y se interpreta como la cantidad de fallas encontradas luego de la entrega del producto. En el caso de CRODA se tomará como el momento actual porque aún no se ha liberado el producto. Según el contexto donde se aplica, se clasifica esta métrica como de producto. Según se infiere lógicamente, esta métrica incrementa su influencia negativa sobre el producto a medida que aumenta su valor cuantitativo.
- ❖ Errores (ERROR): La métrica ERROR está clasificada según su composición como métrica base, se obtiene directamente y el encargado de brindar la información es el jefe de proyecto. El valor obtenido se expresa en unidades y se interpreta como el número de errores encontrados durante las pruebas realizadas al producto. Según el contexto donde se aplica se clasifica esta métrica como de proceso. El valor arrojado debe ser lo más pequeño posible, puede incluso llegar a cero, lo cual sería lo ideal.
- ❖ Índice de Madurez del Software (IMS): La métrica IMS está clasificada según su composición como métrica derivada y corresponde al atributo *madurez*, para su aplicación son usadas otras métricas como el Número Total de Módulos en la Versión Actual (MT), el Número de Módulos en la Versión Actual que se han Cambiado (Fc), el Número de Módulos en la Versión Actual que se han Añadido (Fa) y el Número de Módulos en la Versión Actual que se han Eliminado (Fe). El valor obtenido se expresa en porciento y representa el nivel de estabilidad del proceso. Según el contexto donde se aplica se clasifica esta métrica se clasifica como de proceso. Mientras más se aproxime el IMS al valor unitario (1) más estabilizado estará el producto.
- ❖ Grado de Solución ante Fallos Totales (X): La métrica X está clasificada según su composición como métrica derivada y corresponde al atributo *madurez* del software, para su aplicación son usadas otras métricas como el Número de Fallos Totales Solucionados y el Número Total de Problemas Reales Detectados. El valor obtenido se expresa en porciento y se interpreta como el nivel en que fueron solucionadas las fallas detectadas en el producto. Según el contexto donde se aplica se clasifica esta métrica como de producto. Mientras más se aproxime al valor unitario (1) más

Capítulo I. Fundamentación Teórica

estabilizado estará el producto, porque significaría que fueron solucionados todos los problemas detectados.

- ❖ **Esfuerzo (ESF):** La métrica ESF está clasificada según su composición como métrica derivada. Para su aplicación son usadas otras métricas como Personal Necesario (PN) y Duración (DUR). El valor obtenido se expresa en personas/meses. Según el contexto donde se aplica se clasifica esta métrica como de proceso.
- ❖ **Productividad-Proceso (PR-C):** La métrica PR-C está clasificada según su composición como métrica derivada. Para su aplicación son usadas otras métricas como la Cantidad de Páginas de Documentación (PDOC) y el Personal Necesario (PN). El PN que aquí se refiere es el empleado en la documentación del proyecto, es decir, los revisores técnicos, analistas, arquitectos de información, arquitectos de software y jefe de proyecto. Valor calculado en páginas/personas. Según el contexto donde se aplica se clasifica esta métrica como de proceso.
- ❖ **Productividad-Proyecto (PR-Y):** La métrica PR-Y está clasificada según su composición como métrica derivada. Para su aplicación son usadas otras métricas como la Cantidad en Miles de Líneas de Código Fuente (KLCF) y el Personal Necesario (PN). El PN que aquí se refiere es el empleado en la codificación del proyecto, es decir, los desarrolladores. Valor calculado en cantidad en miles de líneas de código fuente/personas. Según el contexto donde se aplica se clasifica esta métrica como de proyecto.

En la siguiente tabla se relacionan los atributos a evaluar en esta investigación y las métricas correspondientes que permitirán obtener el valor cuantitativo para cada atributo, junto a su clasificación según contexto y composición y su fórmula.

Tabla 1. Relación entre atributos y métricas a aplicar en el proyecto CRODA 2.0

Atributo	Métrica	Clasificación de métrica	Fórmula
----------	---------	--------------------------	---------

Capítulo I. Fundamentación Teórica

Personal	Personal Necesario (PN)	Métrica base ³ . Métrica de proyecto.	-
Duración	Tiempo de Desarrollo Real (TDESR)	Métrica base. Métrica de proyecto.	
Costo	Costo (C)	Métrica base. Métrica de proyecto.	-
Tamaño	Tamaño (T)	Métrica base. Métrica de producto.	-
Defectos	Defectos (DEFCT)	Métrica base. Métrica de producto.	-
Errores	Errores (ERROR)	Métrica base. Métrica de proceso.	-
Madurez	Índice de Madurez del Software (IMS)	Métrica derivada. Métrica de proceso.	$IMS = [MT - (Fc + Fa + Fe)] / MT$ <p>MT: Número de módulos en la versión.</p> <p>Fc: Número de módulos en la versión actual que se han cambiado.</p> <p>Fa: Número de módulos en la versión actual que</p>

³ Las métricas base se obtienen directamente del atributo correspondiente, por tanto no poseen fórmulas de cálculo.

Capítulo I. Fundamentación Teórica

			se han añadido. Fe: Número de módulos en la versión actual que se han eliminado.
	Grado de Solución ante Fallos Totales (X)	Métrica derivada. Métrica de producto.	$X = A_1 / A_2$ A ₁ : Número de fallos totales solucionados. A ₂ : Número total de problemas reales detectados.
Esfuerzo	Esfuerzo (ESF)	Métrica derivada. Métrica de proyecto.	$ESF = PN * DUR$ DUR: Duración del proyecto.
Productividad	Productividad-Proyecto (PR-Y)	Métrica derivada. Métrica de proyecto.	$PR = KLCF / PN^4$ KLCF: Miles de Líneas de Código Fuente.
	Productividad-Proceso (PR-C)	Métrica derivada. Métrica de proceso.	$PR = PDOC / PN^5$ PDOC: Páginas de Documentación.

⁴ Este personal necesario se refiere a los desarrolladores del proyecto, que son los encargados de la codificación.

⁵ Este personal necesario se refiere a los documentadores (analistas y jefe de proyecto), que son los encargados de la documentación.

Capítulo I. Fundamentación Teórica

Con el objetivo esencial de contar con una base que ayude al jefe del proyecto a tomar mejores decisiones con respecto al manejo de los atributos identificados como principales en el proyecto, se propone en esta investigación un procedimiento, que constituirá una guía de apoyo en la toma de decisiones. El procedimiento posibilitará aumentar el nivel de organización, orientación y calidad en el proyecto.

1.6 PROCEDIMIENTO

Se entiende como procedimiento, al conjunto de pasos, claramente definidos, que permiten trabajar correctamente y disminuir la posibilidad de fallos de un producto determinado (Espronceda Jorge y otros, 2010).

Estos pasos son aprendidos y registrados de experiencias pasadas que se repiten para alcanzar las etapas finales de un proceso o producto determinado. Los procedimientos se crean por y para cada institución, acorde a las características propias de cada una de ellas y deben en todo momento estar en correspondencia con las regulaciones establecidas por organismos superiores. Las acciones que conforman los procedimientos, se dirigen a la obtención de una meta, son realizadas para llegar a un resultado. (Suárez Sánchez y otros, 2010)

La elaboración de un procedimiento se logra mediante la recolección de datos relevantes en la institución donde será aplicado y cuenta con la asesoría de personas encargadas de proporcionar las técnicas para el logro. El Procedimiento para evaluar atributos internos de software que será aplicado en CRODA, tiene como objetivo fundamental, proveer una base que ayude al jefe de proyecto a mejorar en la toma de decisiones con respecto al manejo de los atributos identificados como principales en el proyecto. Estas decisiones podrían ser:

- ❖ Retrasar la fecha de entrega del producto.
- ❖ Acortar los gastos del proyecto.
- ❖ Eliminar los errores identificados.
- ❖ Aumentar el nivel de trabajo del personal empleado en el proyecto.

Capítulo I. Fundamentación Teórica

- ❖ Incrementar el nivel de esfuerzo del personal empleado en el proyecto.

Para darle cumplimiento, se tomará como base la evaluación de cada uno de los atributos, que será facilitada mediante la aplicación de las métricas correspondientes para cada uno de ellos. Logrando el manejo adecuado de cada atributo de CRODA, se garantiza la calidad del proyecto en general.

En la bibliografía consultada, fue posible constatar que cada autor de procedimientos, define la estructura de estos, según su criterio y en función de sus necesidades. Por tanto, en la presente investigación se considera que el Procedimiento para evaluar atributos internos de software, para el proyecto CRODA, debe contar con tres fases: Inicio, Medición y Resultados. Estas se componen por 3 flujos, que son enunciados en el orden de la fase correspondiente: Solicitud de evaluación, Medición y Evaluación. En estos flujos se llevan a cabo las siguientes actividades:

- ❖ Confeccionar la planilla Solicitud de Evaluación.
- ❖ Seleccionar las métricas.
- ❖ Aplicar las métricas.
- ❖ Evaluar atributos.
- ❖ Evaluar proyecto.
- ❖ Proponer decisiones.

A raíz del programa de mejora que se lleva a cabo en la universidad fueron modificados algunos artefactos como parte de los cambios en el Expediente de Proyecto del Programa de Mejora. Una de estas modificaciones es la eliminación del Plan de Mediciones, que facilitaba la medición y análisis de los atributos considerados por el jefe de proyecto como principales⁶. Esta modificación fue realizada debido a una de las metas que persigue el

⁶ Contando con la información de estos atributos se satisface la necesidad informativa del proyecto en su mayoría.

Capítulo I. Fundamentación Teórica

proceso de mejora: homogeneizar la información⁷ de los proyectos de software. El Plan de Mediciones no cumplía con esta meta ya que no garantizaba que los atributos medidos fueran los mismos para todos los proyectos que utilizaban el artefacto. En CRODA 2.0 se seguirá el programa de mejora de la siguiente manera: no se utilizará un Plan de Mediciones, en sustitución el proyecto contará con un Procedimiento para evaluar atributos internos de software. Este garantiza la aplicación de las métricas cuya importancia destaca el programa. El procedimiento que se presenta en este trabajo de investigación está dirigido a satisfacer una necesidad puntual y emergente del proyecto y a responder una demanda de información del jefe del proyecto CRODA. En este se miden y analizan los atributos identificados como principales por el jefe del proyecto. Estos permite evaluar los atributos, y a su vez el proyecto en general. Contar con estas evaluaciones posibilita fundamentar un grupo de decisiones que serán propuestas al líder del proyecto. El procedimiento proveerá al jefe de proyecto un Informe de Resultados, este artefacto resumirá los resultados del procedimiento, facilitando la consulta de los mismos. Para adaptar el Procedimiento para evaluar atributos internos de software, al Programa de Mejora, solo es necesario fijar los atributos internos a evaluar y las métricas que serán aplicadas. Lo expuesto anteriormente muestra las ventajas, enfatizando en la flexibilidad, del Procedimiento para evaluar atributos internos de software.

1.7 CONCLUSIONES DEL CAPÍTULO

Luego de haber estudiado y analizado la teoría que da base a la propuesta de solución, se ha arribado a las siguientes conclusiones:

- ❖ El análisis de los atributos internos de software existentes, posibilitó conformar las bases para identificar los atributos principales para CRODA: *personal, duración, costo, tamaño, defectos, errores, madurez, esfuerzo y productividad.*
- ❖ El estudio sobre las métricas de software, permitió seleccionar las correspondientes para los atributos identificados como principales por el jefe del proyecto CRODA. La selección es la siguiente: Personal Necesario (PN), Duración (DUR), Costo (C), Tamaño (T), Defectos (DEFCT), Errores (ERROR), Índice de Madurez del Software

⁷ Información de los atributos de software.

Capítulo I. Fundamentación Teórica

(IMS), Grado de Solución ante Fallos Totales (X), Esfuerzo (ESF), Productividad en Proyecto (PR-Y), Productividad en Proceso (PR-C).

- ❖ El proceso de mejora es un método que ofrece una vía consistente para incrementar el nivel de la calidad de los productos de software. Tomando como base este proceso, para la versión actual de CRODA, no se cuenta con un Plan de Mediciones, para sustituirlo se propone la aplicación de un Procedimiento para evaluar atributos internos de software.

Capítulo II. Propuesta de Procedimiento para evaluar atributos internos de software

CAPÍTULO II. PROPUESTA DE PROCEDIMIENTO PARA EVALUAR ATRIBUTOS INTERNOS DE SOFTWARE

2.1 INTRODUCCIÓN

En este capítulo se describe la propuesta de solución al problema de esta investigación: Procedimiento para evaluar atributos internos de software. Dicha descripción permitirá conocer a fondo aspectos del procedimiento en general como su definición, estructura, sus fases y flujos y artefactos de entrada y de salida. Este procedimiento constituye una guía de apoyo en la toma de decisiones con respecto al manejo de los atributos, lo cual conllevará a mejorar el estado del mismo, tomando como base la teoría estudiada en el capítulo anterior.

2.2 PROCEDIMIENTO. DEFINICIÓN

El procedimiento se encuentra definido de la siguiente manera:

Nombre del procedimiento

Procedimiento para evaluar atributos internos de software.

Objetivo

El objetivo principal del procedimiento para evaluar atributos internos de software es proveer las bases necesarias para incrementar el nivel de calidad de la toma de decisiones.

Alcance

El alcance se define de acuerdo al proyecto donde será aplicado el Procedimiento para evaluar atributos internos de software

2.3 PROCEDIMIENTO. ESTRUCTURA

Capítulo II. Propuesta de Procedimiento para evaluar atributos internos de software

El procedimiento está conformado por tres fases: Inicio, Medición y Resultado, estas a su vez están compuestas por flujos, donde son generadas actividades que son llevadas a cabo por los responsables establecidos

Fase Inicio: Mediante un estudio previo de los atributos internos de software, se procede a identificar los que constituyen de mayor importancia para el proyecto, dada la información que aportan. Para ello se utiliza el método entrevista a los roles que están más capacitados para definirlos. En este caso resulta ser el líder de proyecto.

Flujo 1: Solicitud de evaluación: En el flujo se confecciona una planilla mediante la cual se solicita la aplicación del Procedimiento para evaluar atributos internos de software. Dentro del flujo se desarrolla la siguiente **actividad**:

1. **Confeccionar la planilla Solicitud de Evaluación:** En esta actividad se identifican los atributos internos de software que son necesarios evaluar y se expone el porqué de la esta necesidad.

Responsable: Revisor técnico.

Fase Medición: En esta fase se seleccionan las métricas correspondientes a los atributos que serán evaluados y se aplican las mismas. Cada una de las métricas seleccionadas posibilitará la obtención de un valor numérico para el atributo de software correspondiente. Existen algunos atributos a los que les corresponde más de una métrica, en estos casos, los resultados obtenidos mediante la aplicación de las métricas correspondientes serán promediados, para lograr un valor único que brinde una evaluación cualitativa del proyecto en general.

Flujo 2: Medición: En el flujo se miden los atributos expuestos en la Solicitud de Evaluación. Dentro del flujo definición de las alternativas se desarrolla las siguientes **actividades**:

2. **Seleccionar las métricas:** En esta actividad se seleccionan las métricas a aplicar de acuerdo a los atributos expuestos en la Solicitud de Evaluación.

Capítulo II. Propuesta de Procedimiento para evaluar atributos internos de software

3. **Aplicar las métricas:** En esta actividad se aplican las métricas de software seleccionadas, para obtener los valores cuantitativos correspondientes.

Responsable: Revisor Técnico

Fase Resultados: En esta fase es aplicado el método estudio de caso, en su variante múltiple, para evaluar cada atributo de manera independiente. Con las evaluaciones obtenidas, se procede a evaluar el proyecto en general y a proponer las decisiones con base en las mismas.

Flujo 3: Evaluación: En este flujo se evalúan los atributos expuestos en la Solicitud de Evaluación, en Bien, Regular o Mal. Esto permitirá evaluar el proyecto de manera general y proponer las decisiones basadas en estas evaluaciones. Dentro del flujo definición de las alternativas se desarrolla la siguiente **actividad**:

4. **Evaluar atributos:** En esta actividad se analizan e interpretan los valores obtenidos mediante la aplicación de métricas. Producto a esto se evalúa el estado de los atributos en Bien, Regular o Mal.
5. **Evaluar el proyecto:** Las evaluaciones cualitativas de los atributos serán convertidas en cuantitativas de la siguiente manera:
 - Bien-3
 - Regular-2
 - Mal-1

Esta evaluación cuantitativa será ubicada en una escala, la que se muestra en la siguiente figura. Su valor mínimo es uno y el máximo tres. Si la evaluación cuantitativa se acerca a uno, el proyecto será evaluado cualitativamente de Mal, si se aproxima a dos, Regular y si se acerca a tres, Bien.

Capítulo II. Propuesta de Procedimiento para evaluar atributos internos de software

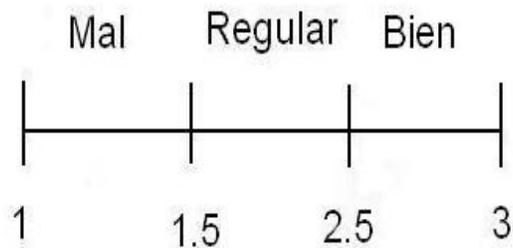


Figura 1. Escala para evaluar el estado del proyecto.

6. **Proponer decisiones:** Al contar con las evaluaciones de cada atributo de manera independiente, y la del proyecto en general, se pueden ser propuestas las decisiones con base en estas evaluaciones.

Responsable: Revisor Técnico

2. 4 PROCEDIMIENTO. ENTRADAS Y SALIDAS

A continuación se esquematizan las entradas y salidas del Procedimiento para evaluar atributos internos de software.

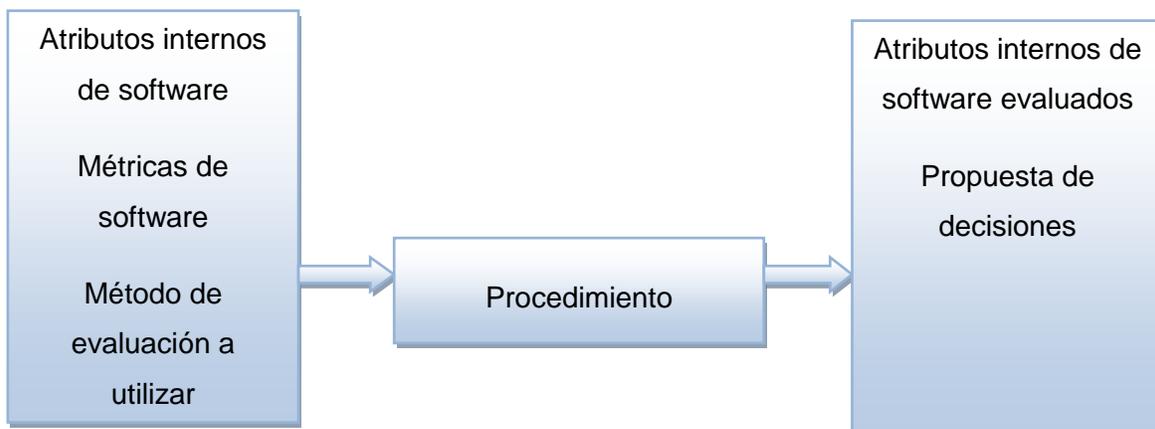


Figura 2. Esquema de entradas y salidas del procedimiento para evaluar atributos internos de software

Las entradas son:

- ❖ Atributos internos de software que serán evaluados.

Capítulo II. Propuesta de Procedimiento para evaluar atributos internos de software

- ❖ Métricas de software correspondientes.
- ❖ Información real y actualizada sobre algunos proyectos de software para aplicar el método estudio de caso

Las salidas son:

- ❖ Evaluación de cada atributo y del proyecto en general.
- ❖ Propuesta de decisiones.

El procedimiento para evaluar atributos internos de software posibilita contar con Artefactos de Entrada y Salida como son:

- ❖ Artefacto de Entrada. Planilla de Solicitud de la Evaluación: En esta planilla debe quedar reflejados varios aspectos como:
 - La situación existente en el proyecto, por la cual se solicita la aplicación del procedimiento.
 - Los atributos que serán evaluados, junto a la fundamentación de por qué estos y no otros.
- ❖ Artefacto de Salida. Informe de Resultados: Este artefacto es elaborado por el revisor técnico, en este se exponen aspectos como:
 - El resumen de las evaluaciones de los atributos y el proyecto en general
 - Una propuesta de las decisiones derivadas de las evaluaciones obtenidas.

2. 5 CONCLUSIONES DEL CAPÍTULO

En este capítulo fue elaborada la propuesta del Procedimiento para evaluar atributos internos de software. Esta permitió establecer el grado de importancia que presenta el mismo, para fundamentar la toma de decisiones, que posibiliten incrementar la calidad del producto final. El Procedimiento para evaluar atributos internos de software quedó estructurado en tres fases: Inicio, Medición y Resultados. Estas se componen por 3 flujos,

Capítulo II. Propuesta de Procedimiento para evaluar atributos internos de software

que son enunciados en el orden de la fase correspondiente: Solicitud de evaluación, Medición y Evaluación. En estos flujos se llevan a cabo las siguientes actividades:

- ❖ Confeccionar la planilla Solicitud de Evaluación.
- ❖ Seleccionar las métricas.
- ❖ Aplicar las métricas.
- ❖ Evaluar atributos.
- ❖ Evaluar proyecto.
- ❖ Proponer decisiones.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

CAPÍTULO III. VALIDACIÓN DEL PROCEDIMIENTO PARA EVALUAR ATRIBUTOS INTERNOS DE SOFTWARE APLICADO A CRODA

3.1 INTRODUCCIÓN

En el presente capítulo valida el procedimiento propuesto mediante su aplicación en el proyecto CRODA. El mismo se valida con los valores reales de los atributos identificados como principales en el proyecto. Con el procedimiento aplicado a CRODA, el jefe de proyecto tendrá una guía con una base real y consistente, que le permitirá mejorar la toma de decisiones con respecto a los atributos.

3.2 RESULTADOS DE LAS MÉTRICAS CORRESPONDIENTES A LOS ATRIBUTOS

Toda la información necesaria para la aplicación de las métricas, fue obtenida mediante entrevistas realizadas al líder del proyecto CRODA. Esto permitió la obtención de los valores cuantitativos que se exponen en la siguiente tabla:

Tabla 2. Valores de los atributos obtenidos mediante métricas

Atributo	Métrica correspondiente	Valor Obtenido
Personal (personas)	Personal Necesario (PN)	20
Duración(meses)	Duración (DUR)	12
Costo(pesos)	Costo (C)	48124.42
Tamaño (unidades de	Tamaño (T)	70

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

casos de uso)		
Defectos(unidades)	Defectos (DEFCT)	0 (aún no se han realizado pruebas del producto)
Errores(unidades)	Errores (ERROR)	20 ⁸
Madurez (porciento)	Índice de Madurez del Software (IMS) Fórmula: IMS = [MT-(Fc+Fa+Fe)]/MT	0
	Grado de Solución ante Fallos Totales (X) Fórmula: X = A1 / A2	60
	Promedio ⁹	30
Esfuerzo(hombres-meses)	Esfuerzo (ESF) Fórmula: ESF = PN * DUR	240
Productividad(unidades)	Productividad-Proyecto (PR-Y) Fórmula: (PR-Y) = KLFC/PN	5
	Productividad-Proceso (PR-C) Fórmula: (PR-C) = PDOC/PN	25

⁸ Esta cifra refleja los errores encontrados de programación específicamente.

⁹ Este atributo utiliza más de una métrica para la obtención del valor cuantitativo correspondiente. Por tanto los valores que arroja cada una de las métricas utilizadas, son promediados para unificar el valor.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

	Promedio ¹⁰	15
--	------------------------	----

A partir de cada uno de estos datos se realizará un análisis mediante el método de evaluación cualitativa, estudio de casos en su variante múltiple, que posibilitará evaluar el atributo en Bien, Regular o Mal.

3.3 MÉTODO DE VALIDACIÓN DEL PROCEDIMIENTO

Existen tres tipos de métodos generales destinados a validar (Labrada Guió, 2009):

- ❖ Métodos cuantitativos: Se basan fundamentalmente en criterios económicos.
- ❖ Métodos cualitativos: Se basan en aspectos cualitativos evaluados.
- ❖ Revisión por pares: Se basa en que el juicio del mérito lo dan expertos que trabajan la temática mediante acuerdos.

Esta investigación, dadas sus características se inclina por los métodos de validación cualitativos, porque estos permitirán generar la evaluación de los atributos internos de software identificados como principales por el líder del proyecto, a partir de una serie de proposiciones extraídas de un cuerpo teórico (Martínez Carazo, 2006). Dentro del método de validación cualitativa se pueden identificar distintos métodos, como:

- ❖ Método experto: Participa un grupo de personas con un grado de experiencia del asunto a tratar, denominados expertos, los cuales son consultados reiteradamente y mediante un proceso establecido llegan a conclusiones del tema tratado. Este método se clasifica en:
 - De una sola iteración con un solo intercambio directo de opiniones (Tormenta de ideas).

¹⁰Este atributo utiliza más de una métrica para la obtención del valor cuantitativo correspondiente. Por tanto los valores que arroja cada una de las métricas utilizadas, son promediados.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

- De una sola iteración sin intercambio (Encuestas).
 - Con varias iteraciones e intercambio directo (Paneles, Mesas redondas).
 - Con varias iteraciones sin intercambio directo (Delphi).
- ❖ Método estudio de caso: Es una metodología rigurosa adecuada para investigar fenómenos en los que se busca dar respuesta a cómo y por qué ocurren. Respecto al diseño de la investigación, los estudios de caso (s) pueden ser simples o múltiples, dependiendo del número de casos que se vaya a estudiar. Partiendo de esto se pueden establecer cuatro tipos básicos, dependiendo del número de casos y de los diferentes niveles de análisis:
- El caso único o unidad de análisis.
 - El caso único con unidad principal y una o más subunidades.
 - Los casos múltiples con unidad principal de análisis.
 - Los casos múltiples con unidad principal y una o más subunidades dentro de la principal.

La autora de esta investigación ha seleccionado como método de validación para la propuesta que fue presentada en el capítulo anterior, el método de evaluación cualitativa estudio de caso de tipo múltiple con unidad principal de análisis. Los casos múltiples que serán estudiados son los proyectos: RHODA, Alfaomega y Multisaber-Navegante, la unidad principal de análisis es el proyecto CRODA. Estos proyectos tienen especificidades en común como el centro al que pertenecen (FORTES) y su meta: elaborar productos con fines formativos para la Web.

A los casos múltiples, solamente se emplearon las métricas para obtener los valores cuantitativos correspondientes a los atributos internos de software, estos fueron ajustados de acuerdo a los identificados para CRODA. La adquisición de estos valores permitirá desarrollar el método estudio de caso haciendo participe del mismo la observación y el análisis comparativo. Esto permitirá llegar a conclusiones concretas con respecto a la evaluación de los atributos de CRODA.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

3.3.1 ESTUDIO DE CASO. DISEÑO

El método de evaluación cualitativa estudio de caso propone 4 pasos fundamentales para ser aplicado:

Paso 1. Selección de la muestra: En este paso son seleccionados los casos que serán estudiados. En esta investigación fueron escogidos los siguientes proyectos de software: CRODA, RHODA, Alfaomega y Multisaber-Navegante.

Paso 2. Definición de las unidades de análisis: En este paso son definidas las unidades que serán analizadas mediante el método estudio de caso. Estas pueden ser (Martínez Carazo, 2006):

- ❖ Tipo 1. Un caso único o unidad de análisis.
- ❖ Tipo 2. Un caso único con unidad principal y una o más subunidades¹¹.
- ❖ Tipo 3. Casos múltiples con unidad principal de análisis.
- ❖ Tipo 4. Casos múltiples con unidad principal y una o más subunidades dentro de la principal.

En la siguiente tabla se representan los tipos de unidades de análisis que han sido expuestos anteriormente.

Tabla 3. Unidades de análisis

Unidad	Caso Único	Casos Múltiples
Simple	Tipo 1	Tipo 3
Múltiple	Tipo 2	Tipo 4

Para validar el Procedimiento para evaluar atributos internos de software en CRODA, se definen las unidades de análisis de Tipo 3 o casos múltiples con unidad principal de

¹¹ Una o más subunidades es tomado como unidad múltiple.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

análisis, donde la unidad es CRODA y los casos múltiples que posibilitaran respaldar el análisis son: RHODA, Alfaomega y Multisaber-Navegante.

Paso 3. Recolección de información: En este paso es recolectada la información que fundamenta el análisis. Se comienza por la identificación de los atributos a evaluar mediante la planilla Solicitud de Evaluación que se presenta a continuación:

Planilla de Solicitud de Evaluación

Herramienta de Autor Web CRODA

Control de Versiones

Fecha	Versión	Descripción	Autor
20/06/2011	1	Elaboración de la planilla de Solicitud de Evaluación	Arianna C. Frenes Padilla

Introducción

El documento refleja mediante una entrevista realizada al jefe de proyecto, el grado de necesidad informativa que el proyecto presenta, las situaciones existentes con respecto a esta y los atributos internos que se necesitan evaluar.

Propósito

La Solicitud de Evaluación es el artefacto de entrada del Procedimiento para evaluar atributos internos de software. Su propósito fundamental es proveer la información suficiente para dar cumplimiento a la Fase Inicio del procedimiento.

Alcance

La presente planilla de Solicitud de Evaluación es elaborada para el proyecto CRODA 2.0.

Preguntas

- 1) ¿Cuáles son los principales problemas que presenta CRODA en su segunda versión, relacionados con la toma de decisiones y la calidad del software?

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

Respuesta: En la segunda versión del proyecto CRODA se ha detectado que la información obtenida para garantizar que la toma de decisiones sea la más correcta no es suficiente, no se conoce el estado de los principales atributos del proyecto, existen deficiencias con respecto al control de la calidad del software y no se emplean elementos tangibles en la evaluación de la estabilidad y capacidad del proceso.

a. ¿Qué consecuencias traen consigo los problemas expuestos?

Respuesta: Si no se cuenta con una base consistente, que respalde la toma de decisiones, no se puede garantizar que la misma se desarrolle de la manera más adecuada. Esto puede traer consigo, que los cambios que se propongan en aras de incrementar la calidad de CRODA, no sean efectivos. Para que el producto final de CRODA, cuente con un nivel elevado de calidad, es necesario que se tenga control total de la misma. Las deficiencias detectadas pueden provocar que no se alcance el nivel de calidad de software esperado e incluso la pérdida del grado de prestigio logrado. En cualquier proyecto es necesario contar con información correcta y actualizada. El hecho de no poseerla puede provocar el fracaso del proyecto, así como la obtención de un producto que no es el deseado.

2) ¿Cómo clasificaría la necesidad informativa actual en el proyecto teniendo en cuenta que las posibles clasificaciones son: Baja, Media o Alta?

Respuesta: En el proyecto existe una alta necesidad informativa.

3) Los atributos internos de software son aquellos que pueden ser medidos examinando el proceso, el producto o el proyecto. Teniendo en cuenta lo anterior y la necesidad informativa existente: ¿Qué atributos internos de software usted clasificaría como de prioridad, en cuanto a la información que reportan para el proyecto?

Respuesta: CRODA cuenta solamente con dos revisores técnicos (estudiantes) que son los encargados de realizar las mediciones. Teniendo esto en cuenta se identifican como atributos necesarios de evaluar: *personal, costo, duración, tamaño, defectos, errores, madurez, esfuerzo y productividad*. Estos son identificados como principales para CRODA

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

porque son los más sencillos y a su vez los que mejor satisfacen las necesidades informativas existentes en el proyecto.

Para recopilar la información referente a los valores de los atributos de cada proyecto a estudiar, fueron empleadas las métricas. Los atributos valorados, para todos los proyectos, han sido los identificados como principales por el líder del proyecto CRODA, para homogenizar la información que será analizada, permitiendo obtener una evaluación para estos.

Paso 4. Análisis de la Información. La información obtenida posibilita analizar los atributos identificados como principales por el líder del proyecto CRODA, para evaluar el estado de los mismos. A continuación se presenta este análisis, con la información recolectada que ha sido reflejada en la *Tabla 4: Valores de atributos de distintos proyectos*.

3.4 ANÁLISIS DE LA EVALUACIÓN DE LOS ATRIBUTOS

La siguiente tabla contiene los atributos a evaluar para CRODA, con sus valores cuantitativos correspondientes que han sido obtenidos mediante la aplicación de métricas. También se encuentran reflejados en la tabla los valores de los mismos atributos pero para los proyectos que constituyen los casos que serán estudiados: RHODA, Alfaomega y Multisaber-Navegante. Esta información permitirá generar la evaluación cualitativa de los atributos identificados como principales por el jefe de proyecto de CRODA, mediante la observación y la comparación.

Tabla 4. Valores de atributos de distintos proyectos

Atributos	Proyectos			
	CRODA	RHODA	Alfaomega	Multisaber-Navegante
Personal (personas)	20	27	116	62
Duración	12	12	24	12

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

(meses)				
Costo (pesos)	48124.42	50000	900000	3.400000
Tamaño (casos de uso)	70	74	171	87
Defectos (unidades)	-	45	-	-
Errores (unidades)	20	63	420	+ de 500
Madurez (porcentaje)	0.3	1	0.19	1
Esfuerzo (hombres-meses)	240	324	2784	1032
Productividad (unidades)	15	6.55	1295.69	90.12

A continuación se expone el análisis de los atributos identificados por el jefe del proyecto CRODA como principales. Este análisis está respaldado por el método de evaluación cualitativa estudio de caso, el cual permitirá evaluar cada atributo como satisfactorio (Bien) o insatisfactorio (Regular o Mal).

El atributo *tamaño* se mide en casos de uso, es necesario tener en cuenta la complejidad de los mismos para evaluarlo. Los casos de uso de proyectos como CRODA y RHODA presentan un nivel de complejidad medio. La mayoría de ellos, involucran más de un diseño de interfaces y son necesarias dos o más entidades de bases de datos. Su escenario se clasifica como exitoso cuando tiene entre cuatro y siete pasos y su implementación involucra entre cinco y diez clases¹². En el caso de Alfaomega y Multisaber-Navegante sus casos de uso resultan altamente complejos. Los mismos

¹²Se refiere a programación.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

involucran una interfaz de usuario compleja y son necesarias tres o más entidades de bases de datos. Su escenario se clasifica como exitoso cuando tiene más de siete pasos y su implementación involucra más de diez clases. Esto permite clasificar al atributo *tamaño* tanto de Alfaomega como de Multisaber-Navegante como grande, mientras que el de CRODA y RHODA es mediano. Considerando lo analizado con respecto a la influencia de la complejidad sobre el *tamaño* y teniendo en cuenta que este atributo, específicamente para el proyecto CRODA, cuenta con un valor de 70 CU se considera que debe ser evaluado de Bien.

Según lo enunciado por Pressman en su libro *Ingeniería de software. Un enfoque práctico*, para llevar a cabo un proyecto considerado como pequeño son necesarias menos de 20 personas (Pressman, 2002). Tomando esto en consideración y analizando la *Tabla 4: Valores de atributos de distintos proyectos*, se puede elaborar una escala para establecer un valor adecuado para la cantidad de personas que debe emplearse en un proyecto de software, tomando en cuenta la influencia del *tamaño*, que se ve reflejada en los casos que se estudian.

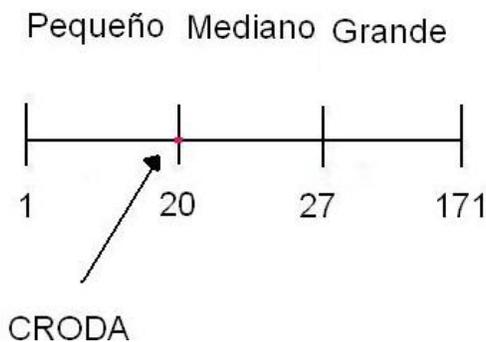


Figura 3. Personal del proyecto CRODA

La escala realizada permite afirmar que mientras mayor sea el *tamaño* (influyendo en este atributo la complejidad del software) del proyecto, más personas serán necesarias para llevarlo a cabo. La cantidad de personas que desarrollan Alfaomega y Multisaber-Navegante, es mucho mayor que las empleadas en CRODA y RHODA, proyectos medianos. El atributo *personal* de CRODA cuenta con un valor cuantitativo de 20

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

personas, de ellas cuatro son analistas. Estos miembros del *personal* son los más relacionados con el valor con que cuenta el atributo *tamaño*, porque este se mide en casos de uso y los analistas, precisamente son los encargados de trabajar con ellos. Según la cantidad de analistas, corresponden por cada uno de ellos aproximadamente 17 casos de uso de mediana complejidad, por tanto se considera que la cantidad de personas para el proyecto CRODA es suficiente. Teniendo en cuenta el análisis expuesto se evalúa al atributo *personal* del proyecto CRODA, con un valor de 20 personas, de Bien.

En la tabla expuesta anteriormente se observa que todos los proyectos sometidos al estudio, excepto Alfaomega, cuentan con el mismo valor para el atributo *duración*, esto se debe al cliente. Para los proyectos cuyo contrato es con Venezuela, se establece, en la etapa de la negociación, un tiempo para la entrega del producto de 12 meses. En el caso específico de Alfaomega, el contrato es con México y es por 96 meses, hasta el momento ha consumido 24 meses de ese tiempo. CRODA hasta la fecha no ha presentado retraso según el cronograma de tiempo establecido. Teniendo en cuenta los argumentos citados se considera que el atributo *duración* del proyecto CRODA, cuyo valor es 12 meses, debe ser evaluado de Bien

El atributo *costo* es establecido al igual que *duración* en la fase de negociación durante el contrato. Todos los proyectos expuestos muestran valores distintos con respecto al *costo*, en los casos extremos se encuentran Multisaber-Navegante (máximo) con 3.400000 pesos y CRODA (mínimo) con 48124.42 pesos. Estos dos proyectos tienen marcada su diferencia desde el primer atributo que fue analizado: *tamaño*. Tomando como referencia a RHODA, por su similitud con CRODA, se considera que el atributo *costo* para este último, que cuenta con un valor de 48124.42 pesos, debe ser evaluado de Bien porque es mucho menor, aun considerando las diferencias con que cuentan los valores de atributos como *tamaño* y *personal* que son bastante pequeñas por lo cual no tienen gran influencia.

El atributo *esfuerzo* se encuentra estrechamente relacionado con *productividad*, esta relación influye positivamente, cuando a medida que aumenta la *productividad* disminuye el *esfuerzo* realizado. Por esta razón estos atributos no son evaluados por separado. Entre los proyectos Alfaomega y Multisaber-Navegante no está clara la influencia positiva

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

de *productividad*–*esfuerzo*, porque la relación necesaria para que esto ocurra y que ha sido expuesta, no se encuentra presente, no siendo el caso para CRODA y RHODA donde esta es bien marcada. En CRODA la *productividad* aumenta hasta en un 56 % con respecto a RHODA y el *esfuerzo* disminuye hasta en un 35%. Con esta base se considera que los atributos *esfuerzo* y *productividad* del proyecto CRODA con valores de 240 y 15 respectivamente, deben ser evaluados de Bien.

El atributo *errores*, que se refiere a las fallas encontradas durante todo el proceso de elaboración del producto, es uno de los atributos que mejora con la disminución de su valor cuantitativo. Si se observan los valores de este atributo para los distintos proyectos estudiados reflejados en la Tabla 4, se puede concluir que con el aumento del *tamaño* se incrementa el número de *errores* encontrados. Los proyectos CRODA y RHODA son considerados como medianos pero entre ellos existe una marcada diferencia con respecto a la cantidad de *errores*. Los detectados en RHODA son más del triple de los que presenta CRODA, 20 *errores*, pero todavía esta cifra se encuentra lejos de la ideal, que es 0. Teniendo esto en cuenta se evalúa el atributo *errores* del proyecto CRODA, con un valor de 20 unidades, como Regular.

El proyecto CRODA no ha sido liberado y tampoco se han realizado pruebas que demuestren que existen *defectos*, que son las fallas que no son detectadas y solucionadas a tiempo y por lo tanto pasan a ser parte del producto que recibe el cliente. En este caso se encuentran también los proyectos Alfaomega y Multisaber–Navegante. En RHODA fueron detectados 45 *defectos*, luego de ser liberado el producto. Por todo lo expuesto se considera que el atributo *defectos* aún no influye en el proyecto.

Para el atributo *madurez* se emplea la métrica Índice de Madurez del Software (IMS), que arrojó un valor de 0. IMS se sitúa en una escala de 0 a 1 y alcanza la mejoría a medida que se aproxima a 1. También para el atributo *madurez* es utilizada la métrica Grado de Solución ante Fallos Totales. Esta tiene como resultado un valor de 0.6, lo que se puede interpretar como un 60 por ciento de fallos totales solucionados. Si se tiene en cuenta solo esto el atributo puede ser evaluado de Regular, pero al promediar el resultado del IMS, ese porcentaje se rebaja a 30. Este valor por encontrarse alejado incluso de la mitad del

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

valor de la *madurez* que se espera alcance el software, trae consigo que el atributo *madurez* sea evaluado de Mal.

Representando este valor en una escala quedaría de la siguiente forma:

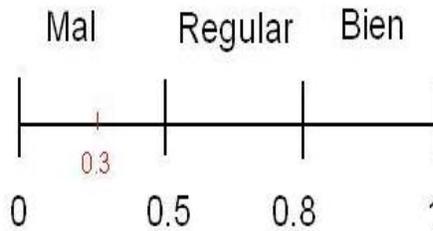


Figura 4. Madurez del proyecto CRODA

Cada una de las evaluaciones de los atributos identificados como principales por el líder del proyecto CRODA, muestra el estado de los mismos. Es necesario disminuir la cantidad de errores detectados, aunque estos no constituyan un riesgo de gravedad para el proyecto, porque podrían convertirse en defectos del producto. Según los valores correspondientes a algunos atributos, se considera que no es obligatorio mejorar los mismos, estos son: *productividad*, *tamaño*, *costo*, *esfuerzo*, *personal* y *duración*. Existe un atributo en particular que dado su valor correspondiente se puede afirmar que es el que influye con más negatividad sobre el proyecto: *madurez*, al contrario de los mencionados anteriormente, mejorar su estado constituye una prioridad para CRODA.

3.5 RESULTADOS DE LA EVALUACIÓN DE CRODA A PARTIR DEL PROCEDIMIENTO PROPUESTO

Como se expone en la tabla que sigue, a cada atributo le será asignado un valor de acuerdo a su evaluación obtenida, si su evaluación es Bien le corresponde 3, si es Regular 2 y si es Mal 1. Estos valores serán promediados y el resultado será llevado a una escala mediante la cual se reflejará el estado del proyecto de manera general.

Tabla 5. Atributos valorados según su evaluación final

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

Atributo	Evaluación Final	Valor
Personal (personas)	Bien	3
Duración (meses)	Bien	3
Costo (pesos)	Bien	3
Tamaño (casos de uso)	Bien	3
Errores (unidades)	Regular	2
Madurez (porciento)	Mal	1
Esfuerzo (meses-hombre)	Bien	3
Productividad(unidades)	Bien	3
Promedio		2.6

Se considera que el resultado del promedio de los valores asignados según las evaluaciones correspondientes a los atributos, debe ser interpretado como el estado del proyecto CRODA. Esto se refleja en la siguiente figura.

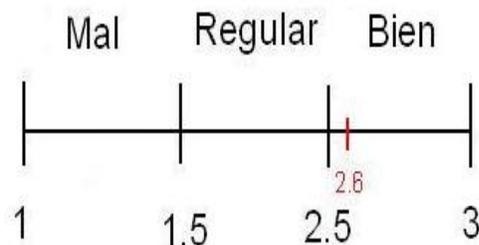


Figura 5. Estado del proyecto CRODA

La información que se brinda como resultado de la validación del Procedimiento para la evaluación de los atributos internos de software en CRODA, constituye para el jefe de proyecto una guía para mejorar la toma de decisiones con respecto a los mismos, con esto no solamente se incrementará el nivel de calidad de cada atributo en particular, aumentará también la calidad del proyecto CRODA en general. Según el resultado de la evaluación de los atributos identificados como principales por el jefe del proyecto CRODA, y la evaluación del proyecto en general, el jefe de proyecto debe tomar las siguientes decisiones, en aras de incrementar la calidad del mismo.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

- ❖ Eliminar los errores identificados.
- ❖ Incrementar el nivel de fallas solucionadas.
- ❖ Disminuir los cambios sobre los módulos realizados, para mejorar el índice de madurez del software conjuntamente con la estabilidad del producto.

Los resultados finales son recopilados en un Informe de Resultados, como se muestra a continuación:

Informe de Resultados

Herramienta de Autor Web CRODA

Control de versiones

Fecha	Versión	Descripción	Autor
20/06/2011	1	Elaboración del Informe de Resultados	Arianna C. Frenes Padilla

Introducción

El documento refleja un resumen de los resultados de la aplicación del Procedimiento para evaluar atributos internos de software en el proyecto CRODA.

Propósito

El Informe de Resultados es el Artefacto de Salida del Procedimiento para evaluar atributos internos de software. Su propósito fundamental es proveer al jefe de proyecto un resumen de los resultados de la aplicación del Procedimiento para evaluar atributos internos de software, en el proyecto.

Alcance

El presente Informe de Resultados es elaborado para el proyecto CRODA 2.0.

1. Medición

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

A continuación se muestra una relación de los atributos a evaluar que fueron identificados en la planilla Solicitud de Evaluación, con el grupo de métricas correspondientes, que fueron seleccionadas y los valores obtenidos de su aplicación. Con esto se da cumplimiento a la Fase Medición del Procedimiento para evaluar atributos internos de software, aplicado a CRODA.

Tabla 6. Medición de atributos

Atributo	Métrica	Fórmula para Aplicación	Valor Obtenido
Personal	Personal Necesario (PN)	–	20
Duración	Tiempo de Desarrollo Real (TDESR)		12
Costo	Costo (C)	–	48124.42
Tamaño	Tamaño (T)	–	70
Defectos	Defectos (DEFCT)	–	0 (aún no se han realizado pruebas del producto)
Errores	Errores (ERROR)	–	20 ¹³

¹³ Esta cifra refleja los errores encontrados de programación específicamente.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

Madurez	Índice de Madurez del Software (IMS)	$IMS = \frac{MT - (Fc + Fa + Fe)}{MT}$ <p>MT: Número de módulos en la versión.</p> <p>Fc: Número de módulos en la versión actual que se han cambiado.</p> <p>Fa: Número de módulos en la versión actual que se han añadido.</p> <p>Fe: Número de módulos en la versión actual que se han eliminado.</p>	0
	Grado de Solución ante Fallos Totales (X)	$X = A1 / A2$ <p>A1: Número de fallos totales solucionados.</p> <p>A2: Número total de problemas reales detectados.</p>	60
Esfuerzo	Esfuerzo (ESF)	$ESF = PN * DUR$ <p>DUR: Duración del proyecto.</p>	240

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

Productividad	Productividad-Proyecto (PR-Y)	PR = KLCF / PN ¹⁴ KLCF: Miles de Líneas de Código Fuente.	5
	Productividad-Proceso (PR-C)	PR = PDOC / PN ¹⁵ PDOC: Páginas de Documentación.	25

En la tabla se observa que existen métricas que no cuentan con fórmula para ser aplicadas, la razón de esto es que están clasificadas según su composición como métricas bases. Las que poseen fórmulas para su aplicación según su composición son clasificadas como métricas derivadas. Los valores correspondientes a las métricas bases son obtenidos directamente y el encargado de esto es el revisor técnico mediante la realización de entrevistas al jefe de proyecto.

Los valores reflejados en la tabla dan paso a la evaluación de los atributos mediante el método estudio de caso, en su variante múltiple.

2. Resultados

A continuación se muestra la aplicación del método estudio de caso, en su variante múltiple, mediante el cual se evalúan cualitativamente los atributos que fueron identificados en la planilla Solicitud de Evaluación. Se tiene como casos múltiples los proyectos: RHODA, Alfaomega y Multisaber-Navegante y como unidad de análisis el proyecto CRODA. La evaluación de cada atributo de manera independiente da paso a

¹⁴ Este personal necesario se refiere a los desarrolladores del proyecto, que son los encargados de la codificación.

¹⁵ Este personal necesario se refiere a los documentadores (analistas y jefe de proyecto), que son los encargados de la documentación.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

evaluar de forma general el proyecto CRODA y a proponer decisiones fundamentadas con las evaluaciones. Con esto se cumple con la Fase 3 y última del Procedimiento para evaluar atributos internos de software, que ha sido aplicado a CRODA.

Tabla 7. Evaluaciones de atributos de CRODA

Atributos	Proyectos				Evaluaciones de atributos de CRODA
	CRODA	RHOD	Alfaomega	Multisaber-Navegante	
Personal (personas)	20	27	116	62	Bien
Duración (meses)	12	12	24	12	Bien
Costo (pesos)	48124.42	50000	900000	3.400000	Bien
Tamaño (casos de uso)	70	74	171	87	Bien
Defectos (unidades)	-	45	-	-	Bien
Errores (unidades)	20	63	420	+ de 500	Regular
Madurez (porcentaje)	0.3	1	0.19	1	Mal
Esfuerzo (hombres-meses)	240	324	2784	1032	Bien
Productividad (unidades)	15	6.55	1295.69	90.12	Bien

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

Para obtener la evaluación cualitativa del proyecto de forma general, a cada atributo le será asignado un valor numérico de acuerdo a su evaluación obtenida, si su evaluación es Bien le corresponde 3, si es Regular 2 y si es Mal 1. Estos valores serán promediados y el resultado será llevado a una escala, que cuenta con el mismo sistema, es decir si el valor promedio se acerca a 3 el proyecto será evaluado de Bien, si se aproxima a 2 será Regular y a 1 se evaluaría el proyecto de Mal. El promedio en este caso fue 2.6. En la escala, referida anteriormente, quedaría de la siguiente manera:

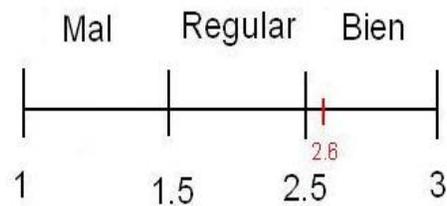


Figura 6. Estado del proyecto CRODA

Según lo observado se evalúa al proyecto CRODA satisfactoriamente (Bien). Estas evaluaciones han dado paso a proponer las siguientes decisiones:

- ❖ Eliminar los errores identificados.
- ❖ Incrementar el nivel de fallas solucionadas.
- ❖ Disminuir los cambios sobre los módulos realizados, para mejorar el índice de madurez del software conjuntamente con la estabilidad del producto.

Capítulo III. Validación del Procedimiento para evaluar atributos internos de software aplicado a CRODA

3.6 CONCLUSIONES DEL CAPÍTULO

Del presente capítulo se concluye que:

- ❖ Se empleó el método de evaluación cualitativa estudio de caso en su variante múltiple, que permitió evaluar los atributos demandados por el jefe del proyecto CRODA mediante el procedimiento.
- ❖ Los atributos fueron evaluados satisfactoriamente (Bien) en su mayoría, excepto por madurez y errores, que fueron evaluados de Mal y Regular respectivamente.
- ❖ El proyecto obtuvo una evaluación satisfactoria, basada en las evaluaciones de los atributos.
- ❖ Se proponen las siguientes decisiones, que se desprenden de las evaluaciones insatisfactorias (Regular y Mal):
 - Erradicar los errores detectados, para evitar que se conviertan en defectos.
 - Incrementar el grado de solución de fallas totales.
 - Disminuir los cambios sobre los módulos para incrementar la estabilidad del producto.

CONCLUSIONES GENERALES

El desarrollo del presente trabajo de diploma permitió dar cumplimiento al objetivo general planteado, al elaborar un Procedimiento para evaluar los atributos internos de software identificados como principales por la dirección del proyecto CRODA. Para obtener dicho resultado, se cumplieron los siguientes aspectos:

- ❖ Con el estudio y análisis de la bibliografía encontrada sobre atributos de software, métricas y procedimientos, se logró asesorar al jefe de proyecto en la identificación de los atributos internos de software con la mayor necesidad de información del proyecto, fueron obtenidos los valores cuantitativos correspondientes a los mismos, mediante la selección y aplicación de métricas y fue creada la base necesaria para proponer un Procedimiento para evaluar atributos internos de software en CRODA.
- ❖ El Procedimiento para evaluar atributos internos de software quedó estructurado en tres fases: Inicio, Medición y Resultados. Estas se componen por 3 flujos, que son enunciados en el orden de la fase correspondiente: Solicitud de evaluación, Medición y Evaluación. En estos flujos se llevan a cabo las siguientes actividades:
 - Confeccionar la planilla Solicitud de Evaluación.
 - Seleccionar las métricas.
 - Aplicar las métricas.
 - Evaluar atributos.
 - Evaluar proyecto.
 - Proponer decisiones.
- ❖ El Procedimiento para evaluar atributos internos de software en CRODA, fue validado mediante el método de evaluación cualitativa estudio de caso. Los atributos fueron evaluados satisfactoriamente (Bien) en su mayoría, excepto por madurez y errores, que fueron evaluados de Mal y Regular respectivamente, esto permitió que el proyecto obtuviera una evaluación satisfactoria, aunque es necesario enfatizar en el deber de

Conclusiones Generales

tomar las decisiones, que se desprenden de las evaluaciones insatisfactorias (Regular y Mal), como:

- Erradicar los errores detectados, para evitar que se conviertan en defectos.
- Incrementar el grado de solución de fallas totales.
- Disminuir los cambios sobre los módulos para incrementar la estabilidad del producto.

RECOMENDACIONES

Luego de cumplir los objetivos propuestos de este Trabajo de Investigación se recomienda para su continuidad:

- ❖ Mejorar el estado de los atributos señalados como insuficientes.
- ❖ Adaptar el Procedimiento para evaluar los atributos internos de software a otros proyectos que no se encuentren incluidos en el Programa de Mejora.

REFERENCIAS BIBLIOGRÁFICAS

Calero Muñoz, Dra. Coral. 2005. *Modelos de Calidad.* [ppt] Castilla-La Mancha: Calidad de Sistemas de Información, 2005.

Calero Muñoz, Dra. Coral. 2006. *Métricas del software: Conceptos básicos, definición y formalización.* [PDF] Castilla - La Mancha, España: Universidad de Castilla - La Mancha, 2006.

Calisoft. 2008. Clasificación de métricas. [Online] 2008.
<http://calisoft.uci.cu/index.php/documentos/18-metrica/58-diagrama-de-la-clasificacion-de-las-metricas-en-las-categorias-de-producto-proyecto-y-proceso>.

Calisoft. 2009. Calisoft (Centro de Calidad para Soluciones Tecnológicas). [Online] 2009.
<http://calisoft.uci.cu/>.

Calisoft. 2010. Calisoft. *Curso de Medición y Análisis.* [Online] marzo 2010.
<http://www.calisoft.uci.cu>.

Calisoft. 2011. *Cambios realizados en el Expediente del Proyecto Versión 3.3.* [PDF] La Habana: Universidad de las Ciencias Informáticas, 2011.

Calisoft. 2011. Evaluación Final del Programa de Mejora. [Online] UCI, mayo 2011.
<http://calisoft.uci.cu/index.php/noticias>.

Centro de Tecnologías para la Formación. 2010. *Centro de Tecnologías para la Formación.* [Documento Visión] La Habana : Centro FORTES, 2010.

Departamento de Lenguajes y Sistemas Informáticos. 2009. Universidad de Sevilla. [Online] 2009. <http://www.lsi.us.es>.

Espronceda Jorge, Katy y León Sosa, Yelina. 2010. *Diseño de un procedimiento para evaluar la confiabilidad de los productos con tecnología Postgres, de DATEC.* [PDF] La Habana: UCI, 2010.

Fuertes Castro, José Luis. 2008. *Calidad del software.* [PDF] La Antigua, Guatemala: Facultad de Informática Universidad Politécnica de Madrid, 2008.

Referencias Bibliográficas

La Torre Hernández, Ludisley y Cepero Núñez, Mariela. 2008. *Métricas de Software*. La Habana, Cuba: s.n., 2008.

Labrada Guió, Lisandra Margarita. 2009. *Procedimiento para evaluar proyectos informáticos y establecer un orden de prioridades*. [PDF] La Habana: UCI, 2009.

Martínez Carazo, Piedad Cristina. 2006. *El método de estudio de caso. Estrategia metodológica de la investigación científica*. [PDF] Barranquilla, Colombia: Universidad del Norte, 2006. ISSN 1657-6276.

Pressman, Roger. 2002. *Ingeniería de software. Un enfoque práctico*. 2002.

Romero, Prof.Dr. Ing. Arturo Luis y Miranda, Lic. Sandor Luis. 2007. Gestipolis.com. [Online] agosto 10, 2007. [Cited: enero 10, 2011.] <http://www.gestipolis.com/administracion-estrategia/la-calidad-historia-conceptos-y-terminos-asociados.htm..>

Serrano, Manuel, y otros. 2005. *Un método para la definición de métricas de software*. [PDF] Castilla – La Mancha, España: Universidad de Castilla – La Mancha, 2005.

Suárez Batista, Ing. Anisbet, y otros. 2011. Informática 2011. [Online] 2011. <http://www.informaticahabana.cu/>.

Suárez Sánchez, Dayli y Silveira Palacio, Yudisleysis . 2010. *Propuesta de Procedimiento para la prestación del servicio de replicación de datos del Centro de Tecnologías y Gestión de Datos*. [PDF] La Habana: UCI, 2010.

Tapanes Alfonso, Maylín. 2008. *Propuesta de métricas para la estimación de proyectos*. [PDF] La Habana: Universidad de las Ciencias Informáticas, 2008.

Universidad de las Ciencias Informáticas. 2010. Redmine del Centro FORTES. [Online] Paquete de Gestion de Proyectos.Direccion Técnica, UCI, 2010. [Cited: enero 23, 2011.] <http://portal.fortes.prod.uci.cu/projects/hacoa>.

Vega Lebrun, Carlos, Rivera Prieto, Laura Susana y García Santillán, Arturo. 2008. *Mejores prácticas para el establecimiento y aseguramiento de la calidad del software*. [PDF] México: Universidad Cristobal Colón, 2008.

BIBLIOGRAFÍA

Alfonso Matos, Yadian y Leyva Sánchez, Leonardo Michel. 2008. *Propuesta de métricas para medir la satisfacción del cliente de los productos en la UCI.* [PDF] La Habana: Universidad de las Ciencias Informáticas, 2008.

Cabrer Rodríguez, Alberto. 2010. *Propuesta de un procedimiento para evaluar la factibilidad de las posibles alternativas de desarrollo de software.* [PDF] La Habana: UCI, 2010.

Calero Muñoz, Dra. Coral. 2005. *Modelos de Calidad.* [ppt] Castilla-La Mancha: Calidad de Sistemas de Información, 2005.

Calero Muñoz, Dra. Coral. 2006. *Métricas del software: Conceptos básicos, definición y formalización.* [PDF] Castilla - La Mancha, España: Universidad de Castilla - La Mancha, 2006.

Calisoft. 2008. Clasificación de métricas. [Online] 2008. <http://calisoft.uci.cu/index.php/documentos/18-metrica/58-diagrama-de-la-clasificacion-de-las-metricas-en-las-categorias-de-producto-proyecto-y-proceso>.

Calisoft. 2009. Calisoft (Centro de Calidad para Soluciones Tecnológicas). [Online] 2009. <http://calisoft.uci.cu/>.

Calisoft. 2010. Calisoft. *Curso de Medición y Análisis.* [Online] marzo 2010. <http://www.calisoft.uci.cu>.

Calisoft. 2011. *Cambios realizados en el Expediente del Proyecto Versión 3.3.* [PDF] La Habana: Universidad de las Ciencias Informáticas, 2011.

Calisoft. 2011. Evaluación Final del Programa de Mejora. [Online] UCI, mayo 2011. <http://calisoft.uci.cu/index.php/noticias>.

Centro de Tecnologías para la Formación. 2010. *Centro de Tecnologías para la Formación.* [Documento Visión] La Habana: Centro FORTES, 2010.

Bibliografía

Departamento de Lenguajes y Sistemas Informáticos. 2009. Universidad de Sevilla. [Online] 2009. <http://www.lsi.us.es>.

Espronceda Jorge, Katy y León Sosa, Yelina. 2010. *Diseño de un procedimiento para evaluar la confiabilidad de los productos con tecnología Postgres, de DATEC.* [PDF] La Habana: UCI, 2010.

Farrat Cardo , Maibel y Wong Leyva,Tecky. 2010. *Procedimiento para el análisis de los resultados de las evaluaciones de software en la UCI.* [PDF] La Habana: UCI, 2010.

Ferreira Aranda, Juan Marcelo, Christian Gómez, Silvano and Rodas, Marcelo. 2008. *CMMI.* [PDF] Paraguay: Universidad Nacional de Asunción, 2008.

Fuertes Castro, José Luis. 2008. *Calidad del software.* [PDF] La Antigua, Guatemala: Facultad de Informática Universidad Politécnica de Madrid, 2008.

La Torre Hernández, Ludisley y Cepero Núñez, Mariela. 2008. *Métricas de Software.* La Habana, Cuba: s.n., 2008.

Labrada Guió, Lisandra Margarita. 2009. *Procedimiento para evaluar proyectos informáticos y establecer un orden de prioridades.* [PDF] La Habana: UCI, 2009.

Martínez Carazo, Piedad Cristina. 2006. *El método de estudio de caso. Estrategia metodológica de la investigación científica.* [PDF] Barranquilla, Colombia: Universidad del Norte, 2006. ISSN 1657-6276.

Morales Sosa, Nairys y González Ginarte, Sergio. 2011. *CMMI, modelo de calidad ideal para realizar la Medición y Análisis en un proyecto.* [PDF] La Habana: Universidad de las Ciencias Informáticas, 2011.

Mulen Mustelier , Evelyn y López Salazar, Yoan. 2010. *Procedimiento para la evaluación de la usabilidad de los productos de software desarrollados en el CESIM.* [PDF] La Habana: UCI, 2010.

Paris Murillo, Carlos Alberto, Damián, Zamorano Martín y Pablo, Manzano Martín. 2008. *CMMI. Calidad de software.* [PDF] México: s.n., 2008.

Bibliografía

Pérez Giraldo, Otoniel. 2006. *Métricas, Estimación y Planificación en Proyectos de Software*. [PDF] Guadalajara, México: Universidad de Guadalajara, 2006.

Pressman, Roger. 2002. *Ingeniería de software. Un enfoque práctico*. 2002.

Pugla Remache, Gabriela Noemí y Leon Quiñonez, Lorena del Cisne. 2008. *Métricas de Proceso y Proyecto*. [PDF] Loja , Ecuador: Universidad Técnica Particular de Loja, 2008.

Quintana Torres , Dayanis Amparo y Cid González, Yusely. 2008. *Propuesta de un Procedimiento para la Toma de Decisiones en la Gestión de Cambios de los Proyectos Productivos de la Facultad 3*. [PDF] La Habana: UCI, 2010

Rigoni Brualla, Cecilia. 2007. *CMMI: mejora del proceso en Fábricas de Software*. [PDF] España : Ministerio de industria, turismo y comercio, 2007.

Rodríguez Tello, Dr. Eduardo A. 2009. *Conceptos básicos de Ingeniería de Software*. [PDF] Tamaulipas, México : CINVESTAV-Centro de Investigación y Estudios Avanzados, 2009.

Romero, Prof.Dr. Ing. Arturo Luis y Miranda, Lic. Sandor Luis. 2007. Gestipolis.com. [Online] agosto 10, 2007. [Cited: enero 10, 2011.] <http://www.gestipolis.com/administracion-estrategia/la-calidad-historia-conceptos-y-terminos-asociados.htm>..

Sálazar Bermudez, Gabriela. 2009. *Midiendo la productividad del desarrollo de software en un curso de Ingeniería de Software: Un caso práctico*. [PDF] Costa Rica : Universidad de Costa Rica, 2009.

Serrano, Manuel, y otros. 2005. *Un método para la definición de métricas de software*. [PDF] Castilla – La Mancha, España : Universidad de Castilla – La Mancha, 2005.

Suárez Batista, Ing. Anisbet, y otros. 2011. *Informática 2011*. [Online] 2011. <http://www.informaticahabana.cu/>.

Bibliografía

Suárez Sánchez, Dayli y Silveira Palacio, Yudisleysis . 2010. *Propuesta de Procedimiento para la prestación del servicio de replicación de datos del Centro de Tecnologías y Gestión de Datos.* [PDF] La Habana : UCI, 2010.

Tapanes Alfonso, Maylín. 2008. *Propuesta de métricas para la estimación de proyectos.* [PDF] La Habana : Universidad de las Ciencias Informáticas, 2008.

Universidad de las Ciencias Informáticas. 2010. Redmine del Centro FORTES. [Online] Paquete de Gestion de Proyectos.Direccion Técnica, UCI, 2010. [Cited: enero 23, 2011.] <http://portal.fortes.prod.uci.cu/projects/hacoa>.

Vega Lebrun, Carlos, Rivera Prieto, Laura Susana y García Santillán, Arturo. 2008. *Mejores prácticas para el establecimiento y aseguramiento de la calidad del software.* [PDF] México : Universidad Cristobal Colón, 2008.

Xitumul Ruiz, Delmi Marilú. 2007. *Normas ISO 9000 VS. CMMI-SW como estándar de calidad en el desarrollo del software y el proceso de la obtención de la certificación en cada estándar.* [PDF] Guatemala : Universidad de San Carlos de Guatemala - Facultad de Ingeniería, 2007.

Yacuzzi, Ing. Enrique. 2005. *El estudio de caso como metodología de investigación: Teoría, mecanismos casuales, validación.* [PDF] Buenos Aires, Argentina : Universidad del Centro de Estudios Macroeconómicos de Argentina - UCEMA, 2005.

GLOSARIO DE TÉRMINOS

Calisoft: La Dirección de Calidad de Software de la Universidad de las Ciencias Informáticas, es un centro de referencia de calidad y órgano de certificación de software, conocido y acreditado a nivel nacional.

CEDIN: Centro de Informática Industrial (CEDIN). Centro generador de soluciones integrales, que desarrolla tecnologías, productos y servicios asociados a la Informática Industrial. Contribuye a la formación especializada y al desarrollo de investigaciones afines que garanticen un alto valor agregado.

CEIGE: Centro de Informatización para la Gestión de Entidades (CEIGE). Centro que se encarga de gestionar una serie de proyectos.

CESIM: Centro de Informática Médica (CESIM). Centro desarrollador de productos, servicios y soluciones informáticas para la optimización del trabajo y mejoramiento de la calidad de la atención médica, contribuyendo a la formación integral de profesionales y permitiendo un posicionamiento en el mercado nacional e internacional.

CMMI: Integración de Modelos de Madurez de Capacidades o Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

Fiabilidad: Muestra hasta dónde se puede esperar que un programa lleve a cabo de su función con la exactitud requerida.

IEEE: Institute of Electrical and Electronics Engineers en español Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), es una asociación técnico-profesional mundial dedicada a la estandarización. Es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, matemáticos aplicados, ingenieros en informática, ingenieros en biomédica, ingenieros en telecomunicación.

Glosario de Términos

ISO 9000: Conjunto de normas sobre calidad y gestión continua de calidad, establecidas por la Organización Internacional de Normalización (ISO). Se pueden aplicar en cualquier tipo de organización o actividad orientada a la producción de bienes o servicios.

ISO 9126: Estándar internacional para la evaluación del Software.

LOM: Learning Object Metadata o Metadatos para Objetos de Aprendizaje (LOM) es un modelo de datos, usado para describir un objeto de aprendizaje y otros recursos digitales similares usados para el apoyo al aprendizaje.

Mantenimiento: El esfuerzo necesario para localizar y arreglar un error en un programa.

OA: Un Objeto de Aprendizaje (OA) es cualquier entidad digital o no digital, que pueda ser usada con fines formativos.

Reusabilidad: (capacidad de reutilización): Muestra hasta dónde se puede volver a emplear un programa (o partes de un programa) en otras aplicaciones.

SCORM: Sharable Content Object Reference Model (SCORM) es una especificación que permite crear objetos pedagógicos estructurados. Con SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM.

SEI: El Software Engineering Institute (SEI) es un instituto federal estadounidense de investigación y desarrollo, fundado por el Congreso de los Estados Unidos en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba al ejército estadounidense la programación e integración de los sub-sistemas de software en la construcción de complejos sistemas militares. Financiado por el Departamento de Defensa de los Estados Unidos y administrado por la Universidad Carnegie Mellon.

TICs: Las Tecnologías para la Información y las Comunicaciones (TICs) agrupan los elementos y las técnicas utilizadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, internet y telecomunicaciones.

Glosario de Términos

Toma de decisiones: La toma de decisiones es el proceso mediante el cual se realiza una elección entre las alternativas o formas para resolver diferentes situaciones, estas se pueden presentar en diferentes contextos: a nivel laboral, familiar, sentimental, empresarial, etc. La toma de decisiones consiste, básicamente, en elegir una alternativa entre las disponibles, a los efectos de resolver un problema actual. En la toma de decisiones importa la elección de un camino a seguir, por lo que en un estado anterior deben evaluarse alternativas de acción. Si estas últimas no están presentes, no existirá decisión.

Universidad Carnegie Mellon: Se ubica en la ciudad de Pittsburgh (Pensilvania) y es uno de los más destacados centros de investigación superior de los Estados Unidos en el área de informática y robótica.

ANEXOS

ANEXO 1. ENCUESTA SOBRE CASOS DE USO DEL PROYECTO

Datos sobre el encuestado

Nombre y apellidos: *<Nombre (s) y apellidos del encuestado>*

Rol: *<Rol que desempeña el encuestado>*

Años de experiencia: *<Años de experiencia desempeñando el rol anteriormente mencionado con que cuenta el encuestado>*

Datos sobre el proyecto

Nombre: *<Nombre del proyecto al que pertenece el encuestado y sobre el que se realiza la encuesta>*

Versión: *<Versión actual del proyecto>*

Centro: *<Centro al que pertenece el proyecto>*

Departamento: *<Departamento al que pertenece el proyecto>*

- 1) Lea detenidamente y responda con cuál de estos casos usted cree que se identifican mayormente los casos de uso realizados en su proyecto. Argumente su respuesta.
 - ❖ Caso 1: El caso de uso usa una interfaz de usuario simple y toca solamente una entidad particular de la base de datos; su escenario exitoso tiene menos de tres pasos; su implementación involucra menos de cinco clases.
 - ❖ Caso 2: El caso de uso involucra más diseño de interfaces y toca dos entidades de bases de datos; su escenario exitoso tiene entre cuatro y siete pasos; su implementación involucra entre cinco y diez clases.
 - ❖ Caso 3: El caso de uso involucra una interfaz de usuario compleja y toca tres o más entidades de bases de datos; su escenario exitoso tiene más de siete pasos; su implementación involucra más de diez clases.

Anexos

- 2) A continuación se presentan varios aspectos relacionados con casos de uso. Identifique los que se corresponden con los casos de uso de su proyecto, marcando con una X en las columnas de Selección.

Tabla 1. Aspectos de Casos de Uso

Aspecto del Caso de Uso	Situación 1	Selección	Situación 2	Selección	Situación 3	Selección
Interfaz que usa	Una simple		Más de una		Una compleja	
Entidad de base de datos relacionada	Una particular		Dos		Tres o más	
Pasos del escenario exitoso	Menos de tres		Entre cuatro y siete pasos		Más de siete pasos	
Clases involucradas en su implementación	Menos de cinco clases		Entre cinco y diez clases		Más de diez clases	

Firma del encuestado (a): _____

Firma del realizador (a) de la encuesta: _____