



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**“ELABORACIÓN Y ANÁLISIS DE MÉTRICAS PARA EL PROCESO DE  
DESARROLLO DE SOFTWARE PARA EMPRESAS DESARROLLADORAS DE  
SOFTWARE DEL ECUADOR”**

## **TESIS DE GRADO**

**Previa a la obtención del Título de:**  
**INGENIERO EN COMPUTACIÓN**  
**ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS**

**Presentado por:**  
**JOSÉ LUIS ALVEAR CERVANTES**  
**JORGE HUMBERTO MAZÓN NARANJO**

**GUAYAQUIL – ECUADOR**

**2007**

# AGRADECIMIENTO

A Dios, porque solo la fortaleza de nuestra fe y las oraciones plegadas han permitido lograr esta meta.

A nuestros padres en especial y familiares por su amor, dedicación y sacrificio brindados a lo largo de nuestras carreras y toda nuestra vida.

A nuestros amigos y demás seres queridos por el apoyo, los consejos y el ánimo positivo compartido especialmente en las situaciones difíciles.

A la ingeniera Mónica Villavicencio, directora de nuestra tesis, por compartir sus conocimientos y hacer posible con su colaboración el desarrollo de la misma.

A nuestros apreciados profesores por colaborar en nuestro desarrollo ético y profesional.

Y de manera muy especial a los profesionales, gerentes y desarrolladores de todas las empresas de software que colaboraron con este trabajo por compartir su valioso tiempo y conocimiento en pro de un mejor futuro de la industria ecuatoriana.

# DEDICATORIA

A Dios.

A nuestros padres.

A nuestros seres queridos.

# TRIBUNAL DE GRADO

---

**ING. HOLGER CEVALLOS**  
Presidente del tribunal

---

**ING. MONICA VILLAVICENCIO**  
Director de tesis

---

**ING. VERONICA UQUILLAS**  
Miembro Principal

---

**ING. VERONICA MACIAS**  
Miembro Principal

# DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

---

**JOSE LUIS ALVEAR C.**

---

**JORGE HUMBERTO MAZON N.**

## RESUMEN

Para resolver el problema de la carencia de información que provenga de la medición de los principales factores que afectan el proceso de desarrollo del software en las empresas del Ecuador dedicadas a esta actividad, se planificó dividir el proyecto de la siguiente forma:

- Primera fase: Se realizó un estudio de tipo exploratorio a las empresas desarrolladoras de software ubicadas en las ciudades de Quito y Guayaquil, para recabar información relevante sobre aspectos de la calidad y de la gestión de los proyectos de software realizados en dichas empresas.
  
- Segunda fase: Se elaboró un plan de medición generalizado y adaptable que sirvió como mecanismo para el registro de los valores cuantitativos y cualitativos de los factores escogidos en base al estudio realizado en la primera fase y que fue aplicado en 3 empresas dedicadas al desarrollo de proyectos de software.

En el Capítulo I se presenta la revisión de investigaciones realizadas anteriormente sobre el tema y la justificación de la tesis resaltando la importancia de las métricas en el desarrollo de software.

El Capítulo II realiza un estudio bibliográfico para analizar conceptos básicos, diferentes modelos de desarrollo, métricas y herramientas para control de calidad del software existentes.

En el Capítulo III se describe el diseño y las fases de la investigación. Los resultados obtenidos sobre las características principales de las empresas de software del Ecuador, el plan de medición de software y las métricas definidas son presentados en el Capítulo IV.

# ÍNDICE GENERAL

	Pág.
RESUMEN.....	VI
INDICE GENERAL.....	VIII
INDICE DE FIGURAS.....	XIV
INDICE DE TABLAS.....	XV
INTRODUCCION.....	XVII
<b>CAPITULO 1: ANTECEDENTES Y JUSTIFICACIÓN.....</b>	<b>1</b>
1.1. El Software.....	2
1.1.1. Importancia en la actualidad.....	2
1.1.2. Evolución a través del tiempo.....	3
1.1.3. La ingeniería de software.....	3
1.2. Estudios previos sobre software ecuatoriano.....	4
1.2.1. Objetivo y procedimiento de estas investigaciones.....	5
1.2.2. Resumen de los principales resultados.....	6
1.2.3. Iniciativas para la mejora del software en otros países.....	10
1.3. Justificación.....	13
1.3.1. Carencia de métricas para el proceso de desarrollo de software.....	13



1.3.2.	Importancia de las métricas para el desarrollo de proyectos de software.....	14
1.3.3.	Factibilidad para realizar la tesis.....	15
<b>CAPITULO 2: MARCO TEÓRICO.....</b>		<b>16</b>
2.1.	La calidad en el software.....	17
2.1.1.	Definición e importancia de la calidad.....	17
2.1.2.	Dificultades para generar productos de calidad.....	18
2.1.3.	El modelo CMMI.....	18
2.2.	Medición y métricas.....	19
2.2.1.	Fundamentos.....	19
2.2.2.	Aplicación de las métricas.....	20
2.2.3.	Guías para el desarrollo de proyectos derivadas del uso de métricas.....	21
2.3.	Procesos de software.....	22
2.3.1.	Definición de proceso.....	22
2.3.2.	Tipos de procesos.....	23
2.4.	Modelos y metodologías para el proceso de desarrollo del software.....	25
2.4.1.	Fundamentos.....	25
2.4.2.	Modelo Cascada.....	25
2.4.3.	Desarrollo evolutivo.....	26

2.4.4.	Desarrollo formal de sistemas.....	26
2.4.5.	Desarrollo basado en la reutilización.....	27
2.4.6.	Iteración de procesos.....	27
2.4.7.	Microsoft Solution Framework.....	29
2.4.8.	Rational Unified Process.....	30
2.4.9.	El proceso de desarrollo de software.....	32
2.5.	Medición del software dirigido a objetivos.....	35
2.5.1.	Fundamentos.....	35
2.5.2.	Aplicación del proceso de medición de software dirigido a objetivos.....	36
2.6.	Herramientas escogidas para la metodología GQIM.....	43
2.6.1.	Técnica nominal de grupos.....	43
2.6.2.	Valor devengado.....	44
2.6.3.	Estadística descriptiva.....	45
2.6.4.	Histogramas.....	46
2.6.5.	Gráficos de Pareto. ....	46
2.6.6.	Gráficos de dispersión.....	47
2.6.7.	Diagrama causa-efecto.....	47
2.6.8.	Complejidad ciclomática.....	48
2.6.9.	Encuestas.....	49

<b>CAPITULO 3: PROCEDIMIENTO</b> .....	51
3.1. Diseño de la investigación.....	52
3.1.1. Tipo de investigación.....	52
3.1.2. Selección del ámbito de estudio.....	53
3.1.3. Fases principales de la investigación.....	53
3.2. Estudio exploratorio.....	55
3.2.1. Formación del instrumento.....	55
3.2.2. Validación y aplicación del instrumento.....	58
3.3. Desarrollo del GQIM.....	59
3.3.1. Objetivos prioritarios de negocio de las empresas de software.....	59
3.3.2. Lista entidad-pregunta.....	60
3.3.3. Sub-Objetivos de negocio.....	64
3.3.4. Entidades y atributos.....	70
3.3.5. Objetivos de medición.....	75
3.3.6. Preguntas cuantificables e indicadores.....	75
3.3.7. Datos a recopilar.....	76
3.3.8. Definición de los datos de las mediciones.....	77
3.3.9. Análisis, diagnóstico y acción.....	79
3.3.10. Plan de medición.....	85
3.4. Aplicación en tres empresas del medio.....	85
3.4.1. Forma de aplicación.....	85

3.4.2.	Facilidades y dificultades en la aplicación.....	86
<b>CAPITULO 4: RESULTADOS.....</b>		<b>88</b>
4.1.	Resumen de las características principales de las empresas de software.....	89
4.1.1.	Aspectos generales sobre las empresas de software.....	89
4.1.2.	Aspectos generales de la gestión de proyectos de software.....	92
4.1.3.	Factores que aumentan el costo del mantenimiento del software.....	103
4.2.	Plan piloto de medición aplicado a 3 empresas de software.....	104
4.2.1.	Características principales.....	105
4.2.2.	Requisitos necesarios para la implementación.....	105
4.2.3.	Limitaciones.....	106
4.2.4.	Recomendaciones para la implementación.....	107
4.3.	Métricas no aplicadas.....	107
4.3.1.	Complejidad ciclomática.....	107
4.3.2.	Conteo de líneas de código.....	108
4.4.	Análisis de las métricas aplicadas durante el plan piloto.....	108
4.4.1.	Satisfacción de los clientes / usuario.....	108
4.4.2.	Esfuerzo del personal.....	110
4.4.3.	Error de estimación.....	111

4.4.4.	Distribución del esfuerzo por fase del proyecto.....	117
4.4.5.	Esfuerzo empleado en documentación.....	119
4.4.6.	Avance del proyecto y complejidad.....	120
4.4.7.	Inspecciones durante el proceso de desarrollo.....	124
4.4.8.	Índice de rendimiento de tiempo y costo.....	125
4.4.9.	Tasa de variación en los requerimientos del software.....	128
4.4.10.	Defectos y fallas.....	129
4.4.11.	Fuga de defectos.....	130
4.4.12.	Distribución por fase del proyecto.....	131
4.4.13.	Distribución por tiempo transcurrido del proyecto.....	132
4.4.14.	Causa – Efecto.....	134
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>136</b>
<b>APENDICE.....</b>		<b>142</b>
A	Cuestionario exploratorio.....	143
B	Plan de medición.....	154
<b>BIBLIOGRAFIA.....</b>		<b>181</b>

## INDICE DE FIGURAS

<b>Figura 2.1.</b> ISO/IEC 12207:1995 Procesos del software.....	24
<b>Figura 3.1.</b> Esquema del cuestionario exploratorio.....	56
<b>Figura 3.2.</b> Factores medidos por las empresas.....	80
<b>Figura 3.3.</b> Documentos elaborados durante el desarrollo de proyectos...	81
<b>Figura 3.4.</b> Formulario F00. Datos generales del proyecto.....	84
<b>Figura 4.1.</b> Años de Funcionamiento.....	91
<b>Figura 4.2.</b> Empresas exportadoras de software.....	92
<b>Figura 4.3.</b> Prioridad de Metas de Desarrollo.....	94
<b>Figura 4.4.</b> Modelamientos de procesos utilizados.....	95
<b>Figura 4.5.</b> Asignación de personal a los proyectos.....	96
<b>Figura 4.6.</b> Distribución del tiempo en los proyectos.....	97
<b>Figura 4.7.</b> Distribución del costo en los proyectos.....	98
<b>Figura 4.8.</b> Consumo de recursos en los proyectos.....	99
<b>Figura 4.9.</b> Problemas con los requerimientos.....	100
<b>Figura 4.10.</b> Tipos de Diseño.....	101
<b>Figura 4.11.</b> Ahorro por Reutilización (tiempo y recursos).....	102
<b>Figura 4.12.</b> Tipos de Pruebas.....	103
<b>Figura 4.13.</b> Factores del costo de mantenimiento.....	104
<b>Figura 4.14.</b> Satisfacción del Cliente/Usuario.....	109

<b>Figura 4.15.</b> Cumplimiento de tareas – Proyecto A.....	113
<b>Figura 4.16.</b> Cumplimiento de tareas – Proyecto B.....	114
<b>Figura 4.17.</b> Cumplimiento de tareas – Proyecto C.....	116
<b>Figura 4.18.</b> Distribución del esfuerzo – Proyecto A.....	117
<b>Figura 4.19.</b> Distribución del esfuerzo – Proyecto B.....	118
<b>Figura 4.20.</b> Distribución del esfuerzo – Proyecto C.....	119
<b>Figura 4.21.</b> Avance del Proyecto A y B.....	121
<b>Figura 4.22.</b> Avance del Proyecto C.....	122
<b>Figura 4.23.</b> Complejidad de los Proyectos.....	123
<b>Figura 4.24.</b> Índices de Rendimiento Proyecto A.....	127
<b>Figura 4.25.</b> Índices de Rendimiento Proyecto B.....	128
<b>Figura 4.26.</b> Distribución de defectos Proyecto A.....	133
<b>Figura 4.27.</b> Distribución de Fallas Proyecto A.....	133
<b>Figura 4.28.</b> Diagrama de Causas – Efectos.....	134

## INDICE DE TABLAS

<b>Tabla 2.1.</b> Medición de software basado en objetivos.....	36
<b>Tabla 3.1.</b> Objetivos del negocio - Ref.....	60
<b>Tabla 3.2.</b> Entradas y recursos.....	60
<b>Tabla 3.3.</b> Actividades y flujos.....	61
<b>Tabla 3.4.</b> Artefactos internos.....	63
<b>Tabla 3.5.</b> Productos y derivados.....	63
<b>Tabla 3.6.</b> Agrupación Nro. 1. Cliente.....	65
<b>Tabla 3.7.</b> Agrupación Nro. 2: Personal, conocimientos y experiencias.....	65
<b>Tabla 3.8.</b> Agrupación Nro. 3: Herramientas.....	66
<b>Tabla 3.9.</b> Agrupación Nro. 4: Documentación.....	66
<b>Tabla 3.10.</b> Agrupación Nro. 5: Proceso de desarrollo del software, (Consumibles, planeación del proyecto, cronograma, presupuesto).....	67
<b>Tabla 3.11.</b> Agrupación Nro. 6: Requerimientos.....	68
<b>Tabla 3.12.</b> Agrupación Nro. 7. Diseño.....	69
<b>Tabla 3.13.</b> Agrupación Nro. 8. Código fuente.....	69
<b>Tabla 3.14.</b> Agrupación Nro. 9. Defectos.....	69
<b>Tabla 3.15.</b> Cliente / Usuario final.....	71
<b>Tabla 3.16.</b> Personal.....	71
<b>Tabla 3.17.</b> Herramientas.....	71



<b>Tabla 3.18.</b> Documentos.....	72
<b>Tabla 3.19.</b> Proceso de desarrollo.....	72
<b>Tabla 3.20.</b> Requerimientos.....	73
<b>Tabla 3.21.</b> Diseño.....	73
<b>Tabla 3.22.</b> Código fuente.....	74
<b>Tabla 3.23.</b> Defectos / Fallas.....	74
<b>Tabla 3.24.</b> Objetivo de medición Nro. 1.....	75
<b>Tabla 3.25.</b> Preguntas cuantificables e indicador.....	76
<b>Tabla 3.26.</b> Datos necesarios para formar el indicador.....	77
<b>Tabla 3.27.</b> Referencia a la definición de los datos a recopilar.....	78
<b>Tabla 3.28.</b> Disponibilidad de Datos.....	83
<b>Tabla 4.1.</b> Ubicación y Tamaño.....	90
<b>Tabla 4.2.</b> Metas de Desarrollo.....	93
<b>Tabla 4.3.</b> Inspecciones durante el proceso.....	124
<b>Tabla 4.4.</b> Cambios de Requerimientos.....	129

## INTRODUCCION

El objetivo de este proyecto de tesis es determinar el uso actual de métricas en las empresas ecuatorianas desarrolladoras de software el mismo que nos servirá como una referencia necesaria para la formación de un plan de métricas utilizando un proceso conocido como “medición del software dirigido a objetivos” propuesto por el Instituto de Ingeniería de Software (SEI) de Universidad Carnegie Mellon con el propósito de utilizarlo para la evaluación, seguimiento y control del proceso de desarrollo de software.

Con la ayuda de la Escuela Superior Politécnica del Litoral y a través del proyecto VLIR-ESPOL, se inició un estudio de tipo exploratorio en 63 empresas desarrolladoras de software ubicadas en las ciudades de Quito y Guayaquil mejorando de esta manera el estado actual de conocimiento sobre el uso de métricas en las empresas. Con la información adquirida se desarrolló el proceso de medición de software dirigido a objetivos definiéndose objetivos de medición, identificándose necesidades de información y construyéndose indicadores que visualizan el estado del proceso de desarrollo del software a medida que este se va desarrollando.

Finalmente se hizo una aplicación piloto del plan de métricas en 3 empresas de software que permitió identificar aspectos favorables y desfavorables que deben ser tomados en cuenta para evitar errores en la aplicación de este instrumento y mejorar la precisión de los resultados obtenidos.

CAPITULO



# Antecedentes y Justificación

# CAPITULO 1

## ANTECEDENTES Y JUSTIFICACIÓN

### 1.1. EL SOFTWARE

A diferencia de la visión restrictiva que tienen muchas personas, el software consiste en diversos programas de computadora independientes junto con todos los documentos asociados y la configuración que se necesita para hacer que estos programas operen de manera correcta para producir, gestionar, adquirir, modificar, mostrar o transmitir información. [1]

#### 1.1.1. Importancia en la actualidad

Casi todas las actividades comerciales y la toma de decisiones gerenciales dependen de sistemas complejos realizados en computadora. Sin un software complejo no habríamos explorado el espacio, no tendríamos Internet y telecomunicaciones modernas, y todas las formas de viajar serían más peligrosas y caras [2].

### 1.1.2. Evolución a través del tiempo

En la evolución del software se han venido dando problemas que continúan en aumento, entre ellos tenemos [2]:

- Los avances en el software continúan dejando atrás la habilidad de construir software para utilizar toda la potencialidad del hardware.
- No se puede construir nuevos programas tan rápido como lo exige el ritmo de la demanda del mercado y del negocio.
- El uso extenso de computadoras ha hecho de la sociedad más dependiente de la operación fiable del software.
- Se lucha por construir software que tenga fiabilidad y alta calidad.
- La capacidad de soporte y la de mejorar los programas existentes se ve amenazada por diseños pobres y recursos inadecuados.

A medida que transcurra el siglo XXI, el software se convertirá en uno de los pilares que conduzca a grandes cambios desde la educación elemental hasta la ingeniería genética [2].

### 1.1.3. La Ingeniería de Software

A finales de los años 60s, era evidente que un enfoque informal para el desarrollo del software no era adecuado debido a los años

de retrasos y exceso en los presupuestos de los proyectos de esa época [1]. Propuesta inicialmente en 1968, en una conferencia para discutir lo que en ese entonces se llamó “La crisis del software”, La Ingeniería de Software se había llegado a definir como “una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza” y cuya meta es el desarrollo costeable de sistemas de software [1].

## **1.2. ESTUDIOS PREVIOS SOBRE SOFTWARE ECUATORIANO**

Entre los estudios previos más importantes que se conocen sobre software ecuatoriano se encuentran: 1) “Estudio del Sector de Tecnología de Información del Ecuador: Software y Servicios” realizado por la Corporación de Promoción de Exportaciones e Inversiones CORPEI”, 2) “Primer Estudio de La Industria de Software Del Ecuador” realizado por la Asociación Ecuatoriana de Software AESOFT, y 3) “Análisis Estadístico Exploratorio de las Empresas Desarrolladoras de Software Asentadas en Guayaquil, Quito y Cuenca” realizado por el Sub-componente 8 de Ingeniería de Software del proyecto VLIR-ESPOL.

### **1.2.1. Objetivo y procedimiento de estas investigaciones**

La CORPEI nace a través de la Ley de Comercio Exterior e Inversiones (LEXI) el 9 de Junio de 1997, como institución privada sin fines de lucro, responsable de la promoción de la imagen, las exportaciones y las inversiones del Ecuador, dentro y fuera del país. El estudio realizado en el 2001 entrega información sobre las exportaciones de software ecuatoriano, productos exportados y la oferta exportable del sector de software ecuatoriano y analiza ampliamente casos de éxito en la exportación de software de algunas empresas, y los nichos en el mercado internacional. **[3]**

La Escuela Superior Politécnica del Litoral (ESPOL) en asociación con el Consorcio de Universidades Flamencas desde Abril del 2003 tiene como uno de sus objetivos brindar apoyo a la industria de software del país. Con el objetivo de conocer las características relacionadas con el uso de estándares de calidad en desarrollo de software se elaboró un cuestionario que se administró a empresas de Quito, Guayaquil y Cuenca en el mes de Octubre del 2003. El cuestionario genera información sobre las características más relevantes de las empresas y sus gerentes de desarrollo **[4]**.



La AESOFT, es una organización gremial privada sin ánimo de lucro creada en mayo de 1995 en Quito, Ecuador. A través del estudio se busca generar una herramienta de gestión para el sector privado, gubernamental y académico que permita incorporar planes a largo plazo en la Industria del software. Desarrollado con colaboración de la Empresa Datanalisis con información de las empresas de software en el año 2003 y 2004 el estudio de la industria del software en el país realizado por AESOFT nos muestra un análisis completo del mercado con cifras estadísticas del perfil de la industria, perfil del recurso humano utilizado, barreras del crecimiento en la industria, ventas al exterior, comercialización, facturación, y perspectivas de desarrollo [5].

### **1.2.2. Resumen de los principales resultados**

El estudio realizado por La CORPEI (2001) a más de 21 empresas, sobre las exportaciones de software ecuatoriano, descubrió que:

- Entre 1990 y 1995 se inició la exportación de software en el Ecuador; para 1999 se estimó que las empresas del sector habían exportado USD 15 millones y las ventas de software ecuatoriano crecieron sostenidamente un 40% cada año

desde 1995 en Latinoamérica y en algunos casos, en Estados Unidos.

- Ecuador poseía una calidad de Recurso Humano de primer orden de alrededor de 1.200 personas.
- La mayoría de empresas tenían un promedio de edad mayor a 5 años y se encontraban en procesos de certificación ISO 9001 y algunas que tenían más de 10 años ya lo habían obtenido.
- Los productos exportados más comunes eran software financiero y bancario, de recurso humano, de lingüística, administrativo entre otros. Por otra parte algunas ya ofrecían servicios de outsourcing de desarrollo de software y de call centers
- Entre las principales dificultades se encontraron que las universidades en Ecuador no poseían capacidad de ofrecer personal especializado para desarrollo de software altamente técnico, las empresas que se encontraban en el proceso de obtención de certificaciones de calidad necesitaban mecanismos accesibles y muy pocos gerentes, directores de proyectos poseían la capacidad para administrar el proceso de exportación de software.

Los principales resultados del estudio aplicado a 77 empresas de software ubicadas en Quito, Guayaquil y Cuenca por el Sub-Componente 8 del proyecto VLIR-ESPOL (2003) fueron los siguientes:

- El 51% de las empresas son medianas o PYMES, el 40.4% son pequeñas y el 8.6% son grandes. El 46% de las empresas realizan consultorías, desarrollo y venta de sus aplicaciones en conjunto.
- La edad promedio de los gerentes bordea los 33 años, y la edad promedio de las empresas es 7.4 años. Sólo el 19,4 % de las empresas utilizan una combinación entre experiencia, habilidades y capacidades de los desarrolladores.
- El mercado objetivo se divide según el tamaño de las mismas y lo conforman las empresas comerciales, de servicios, financieras, industriales y gubernamentales. Sólo el 36,4% penetra en el mercado internacional.
- En el medio sí se tiene conocimiento sobre las normas de calidad en el desarrollo del software, pero al parecer no sobre los beneficios que podrían obtener al estar certificados.

- El 36,3% de las empresas utilizan estándares de calidad en el desarrollo de software de los cuales sólo el 24,6% son internacionalmente reconocidos. La mayoría utiliza procedimientos internos para asegurar la calidad en el software.

El estudio de AESOFT (2003-2004) realizado con una muestra de 55 empresas PYMES y 7 empresas grandes obtuvo que:

- Existen un total de 223 empresas en el país, que generaron ventas por \$62 millones, exportaciones \$10.7 millones y un aporte fiscal \$21.6 millones.
- Número de empleos directos fijos: 2.600 de lo cuales el 35,9% son ocupados por profesionales y técnicos.
- Entre las barreras que enfrenta la industria están: “Disponibilidad de recursos económicos”, y “Disponibilidad de recursos humanos”. Entre los principales problemas que enfrenta la industria se encuentran: “Falta de esquemas de certificación de calidad internacional accesibles” (28.4%), y “Deficiencias en la capacidad gerencial de las empresas” (13.4%). Y una de las estrategias adecuadas para estimular el sector es “Crear mecanismos para el acceso a certificación internacional”.

- Solo el 29.9% de las empresas ha prestado servicio al exterior, siendo las regiones de Latinoamérica y Norteamérica los destinos más comunes.

### **1.2.3. Iniciativas para la mejora del software en otros países**

Varios países de Latinoamérica se encuentran realizando actividades de mejora que permitan a sus respectivas empresas de software mejorar sus capacidades mediante planes de apoyo especialmente dirigidos a lograr la exportación de productos y servicios de software.

En **Brasil**, existe un plan federal de apoyo a la industria del software que aportó \$ 100 millones en 4 años, asesoró a 26 empresas para la certificación de calidad ISO 9001 y realizó 26 evaluaciones SW-CMM hasta agosto del 2004 [6]. Existen Organizaciones como la Asociación de Empresas Brasileñas de Tecnología de Información (ASSESPRO), el Programa Nacional de Software para Exportación (SOFTEX) y el Programa para el desarrollo de la Industria Nacional de Software y Servicios Relacionados (PROSOFT) difundiendo actividades para el mejoramiento de la calidad, inversión y exportación de software [6].

En **México** la Asociación Mexicana de la Industria de Tecnologías de Información (AMITI) impulsa evaluaciones de SW-CMM [6], La Asociación Mexicana para la Calidad en la Ingeniería de Software (AMCIS) ofrece un diplomado en calidad de software desde 2002. Se difunde el Modelo de Procesos para la Industria del Software en México (MoProsoft) para pequeñas y medianas empresas. En ESICenter Mexico se organizan eventos, diplomados capacitación y evaluaciones de estándares de calidad. El Programa para el Desarrollo de la Industria de Software (PROSOFT) promueve inversiones y exportaciones planeando metas para el año 2013.

[6]

En **Chile** la Asociación Chilena de Empresas de Tecnología de Información (ACTI) organiza estudios sobre el tamaño y ventas de las empresas. SPIN-Chile desarrolló un estándar S:PRIME basado en CMM y difunde su uso en proyectos de empresas pequeñas. La Sociedad Chilena de Software y Servicios (GECHS) está iniciando un proceso de certificación internacional ISO 9001-2000 y CMM (nivel 2) con 12 empresas y está elaborando el 3er Estudio de Diagnóstico del Sector Software. La

Universidad de Chile ofrece un Diplomado en Calidad de Software [6].

En **Perú** también hay entidades como la Asociación Peruana de Productores de Software (APESOFT) que realiza estudios sobre el tamaño promedio y las ventas de las empresas de software además de evaluaciones SW-CMM. El gobierno tiene un proyecto de ley que declara ser de necesidad pública e interés nacional el desarrollo de esta industria, encamina la creación de un parque tecnológico para empresas de software y un congreso [6]. El Programa de Apoyo a la Competitividad de la Industria del Software (PACIS) ejecuta asistencia técnica y capacita consultores en mejora de procesos basándose en el modelo CMMI [6].

En **Costa Rica** la Cámara de productores de software (CAPROSOFT) creó el proyecto PRO-SOFTWARE en 1999 para compartir los costos y beneficios con las empresas de software pequeñas en el desarrollo de proyectos de mejoramiento del proceso de software, tienen 2 empresas en SW-CMM nivel 3 y 2 empresas certificadas ISO 9000 [6]. La empresa CENFOTEC ofrece un curso de certificación en calidad de software. [6]

### **1.3. JUSTIFICACIÓN**

La tesis se enfoca en uno de los problemas que surge por la carencia de información proveniente del proceso de desarrollo del software en las empresas del Ecuador, la medición y los resultados obtenidos del proceso de desarrollo en una muestra de la industria del software pueden servir como referencia para las empresas interesadas en hacer mediciones de sus procesos. Adicionalmente esta tesis brinda la oportunidad de mejorar las relaciones academia-industria.

#### **1.3.1. Carencia de métricas para el proceso de desarrollo de software**

Durante la revisión de la bibliografía y de los estudios existentes sobre el software ecuatoriano, la ausencia de datos cuantitativos y cualitativos sobre aspectos que influyen el desarrollo de los proyectos de software fue notoria, razón por la cual el proponer un plan de métricas resulta necesario para la mejor estimación y control de este tipo de proyectos. Las mediciones que se desarrollarán deben estar enfocadas para cumplir con objetivos y metas que permitan mejorar el proceso de desarrollo del software. Para facilitar la recolección de datos estos deben ser registrados en instrumentos adecuados como formularios y



encuestas que sean completos y concisos que puedan ser fácilmente resumidos.

### **1.3.2. Importancia de las métricas para el desarrollo de proyectos de software**

La medición produce cambios culturales. Los ingenieros del software y sus gestores pueden obtener una visión más profunda del trabajo que realizan y del producto que elaboran creando una línea base de métricas –una base de datos que contenga mediciones del proceso y del producto. [7]

Entre las principales ventajas podemos citar que se logra identificar oportunidades de mejoras en el proceso, ayuda a establecer prioridades a los problemas y provee una base para mejorar ordenadamente el esfuerzo [7]. El proceso de software se puede medir para mejorarlo, para ayudar a estimar, controlar la calidad, evaluar la productividad y el control de proyectos. Se mide la calidad del producto para ayudar en la toma de decisiones a medida que el proyecto evoluciona. [7]

### **1.3.3. Factibilidad para realizar la tesis**

Para la realización de esta tesis, se cuenta con el apoyo del Sub-Componente 8 de Ingeniería de Software del proyecto VLIR-ESPOL el cual cuenta con un laboratorio ubicado en la Facultad de Ingeniería en Electricidad y Computación (FIEC) a disposición de los estudiantes de Ingeniería de Software y de los tesisistas cuyos temas tengan relación con la Ingeniería de Software, medios de comunicación telefónica y agendas de teléfonos para contactar a las empresas.

Adicionalmente, se dispone de espacio físico en las oficinas de la ESPOL en Quito, el uso de un computador y medios de comunicación. El proyecto de tesis es solventado por cofinanciamiento obtenido en la participación del concurso "Proyectos Semilla 2005" organizado por el CICYT.

CAPITULO

2

Marco Teórico

# CAPITULO 2

## MARCO TEÓRICO

### 2.1. LA CALIDAD EN EL SOFTWARE

El termino calidad del software es un concepto ambiguo y complejo que resulta difícil de definir, pues existen varios factores que se deben tener en cuenta al momento de considerar la calidad de un producto de software.

#### 2.1.1. Definición e importancia de la calidad

No obstante, a pesar de la dificultad mencionada, una definición útil de “calidad” es expresada por Crosby (1979) como la “Conformidad con los requerimientos” y complementada por Juran y Gryna (1970) como “la conveniencia para el uso”. Estas dos definiciones están relacionadas, son consistentes y han sido adoptadas y utilizadas por muchos profesionales de la calidad [8]. Debido a que el software está presente en la vida diaria de las personas, la calidad resulta ser un factor por demás importante debido a los efectos negativos e incluso graves que pueden tener sobre las personas que dependen o se ven afectadas por la dificultad en su operación, “caídas” del sistema, dificultad en su mantenimiento, etc.

### **2.1.2. Dificultades para generar productos de calidad**

Entre las múltiples causas que dificultan generar productos de calidad tenemos que: La calidad del software no depende de un proceso de manufactura sino de un proceso de diseño donde las consideraciones humanas individuales son importantes. [9] Ciertas características de la calidad son difíciles de expresar de forma no ambigua o incluso no se especifican de forma explícita, por ejemplo: La facilidad de mantenimiento, la eficiencia, etc. [1]. De igual forma resulta difícil redactar especificaciones concretas para el software de forma que aunque un producto de software cumpla con los requerimientos los usuarios no lo consideran un producto de calidad [1].

### **2.1.3. El modelo CMMI**

Desarrollado por el Instituto de Ingeniería de software (SEI) como una forma de convencer a la compañías que desarrollaban software para el departamento de defensa de los Estados Unidos de Norte América a mejorar el software que entregaban [10], el modelo CMMI (Capability Maturity Model Integration) provee de una visión estructurada para la mejora de los procesos en una organización ayudando a establecer objetivos y prioridades para

la mejora de estos procesos, facilita guías para los procesos de calidad y provee un estándar utilizado para la evaluación de prácticas actuales. [11]

## **2.2. MEDICION Y MÉTRICAS**

La medición del software se refiere a derivar un valor numérico para algún atributo de un producto de software o un proceso de software [1], y una métrica de software es cualquier tipo de medida relacionada con un sistema proceso o documentación de software [1].

### **2.2.1. Fundamentos**

Una medición cuantifica una característica de un proceso o de un producto. La medición puede ser directamente cantidades observables o se puede derivar de una o más cantidades directamente observables. Una métrica se define como la relación entre dos medidas. Los típicos ejemplos de métricas incluyen el número de las líneas fuente del código, el número de las páginas de la documentación, el número de horas-staff, el número de pruebas, el número de requerimientos, etc. Los ejemplos de métricas derivadas incluyen líneas del código fuente por horas-staff, defectos por mil líneas de código, o un índice del costo. [12]

Pero para definir una métrica se deben definir primero: el propósito, alcance, atributo a medir, instrumento de medición, escalas y variación del atributo e instrumento, teóricamente la relación de los datos y los probables efectos colaterales. [13]

Los modelos de calidad de un producto de software incluyen a menudo las mediciones para determinar el grado de calidad de cada característica del producto, si se seleccionan apropiadamente, las medidas pueden apoyar la calidad del software y entre otros aspectos el ciclo de vida del software. [14]

Hoy en día la medición es considerada una disciplina clave en la ingeniería de software, como evidencia de ello, esta su inclusión en los requerimientos de madurez del nivel 2 del modelo CMMI [15].

### **2.2.2. Aplicación de las métricas**

Toda organización de software exitosa implementa mediciones como parte de sus actividades técnicas y administrativas diarias. Ya que la medición les provee la información objetiva que necesitan para tomar decisiones que impacten positivamente el rendimiento de sus negocios [15].

La medición produce cambios culturales. Los ingenieros del software y sus gestores pueden obtener una visión más profunda

del trabajo que realizan y del producto que elaboran creando una línea base de métricas – una base de datos que contenga mediciones del proceso y del producto. [7]

La utilidad de la aplicación de las métricas la podemos observar desde dos frentes dentro de una empresa, desde la perspectiva de procesos, productos y recursos del software para caracterizar, evaluar, predecir y mejorar; y desde la administración del desarrollo del software para planear, organizar, controlar y mejorar. [7]

### **2.2.3. Guías para el desarrollo de proyectos derivadas del uso de métricas**

El uso de las métricas ha permitido observar tendencias y establecer “guías” que permiten tener en cuenta la experiencia del desarrollo de proyectos anteriores en la estimación y control de nuevos proyectos de software. Robert B. Grady [10] presenta un compendio de estas útiles guías conocidas también como “reglas del pulgar”, Entre estas tenemos las siguientes:

- Los proyectos creados principalmente de software re-utilizado toman un cuarto del tiempo en su desarrollo comparado a aquellos que son nuevos.



- De 50% a 75% de todos los errores de diseño pueden ser encontrados con inspecciones.
- Se encontrará un defecto después de la liberación del software por cada 10 defectos encontrados durante las pruebas.
- Toma de 4 a 10 veces más tiempo corregir defectos en grandes y modificados o actualizados sistemas de software que hacer arreglos antes o al poco tiempo de la liberación inicial de un sistema.

### **2.3. PROCESOS DE SOFTWARE**

Para resolver los problemas reales de una industria, un ingeniero de software debe incorporar una estrategia de desarrollo que a menudo se llama modelo de proceso o paradigma de Ingeniería de Software y se forma de un conjunto de tareas que pueden ser hitos de proyectos, posibles entregas, una colección de tareas de Ingeniería de Software, productos de trabajo del software y puntos de garantía de calidad. [2]

#### **2.3.1. Definición de proceso**

Es un conjunto de actividades y resultados asociados que producen un producto de software [1]. En un proceso se hace énfasis en “como” se harán las diferentes actividades que lo

conforman, las mismas que suceden en cierto orden o secuencia lógica.

### **2.3.2. Tipos de procesos**

El estándar ISO/IEC 12207:1995 provee un marco común, el cual cubre el ciclo de vida del software desde su conceptualización hasta su retiro, y consiste de procesos para adquirir y suministrar productos y servicios de software. Este marco permite controlar y mejorar los procesos que se aprecian en la figura 2.1.

ISO/IEC 12207:1995 Procesos del software

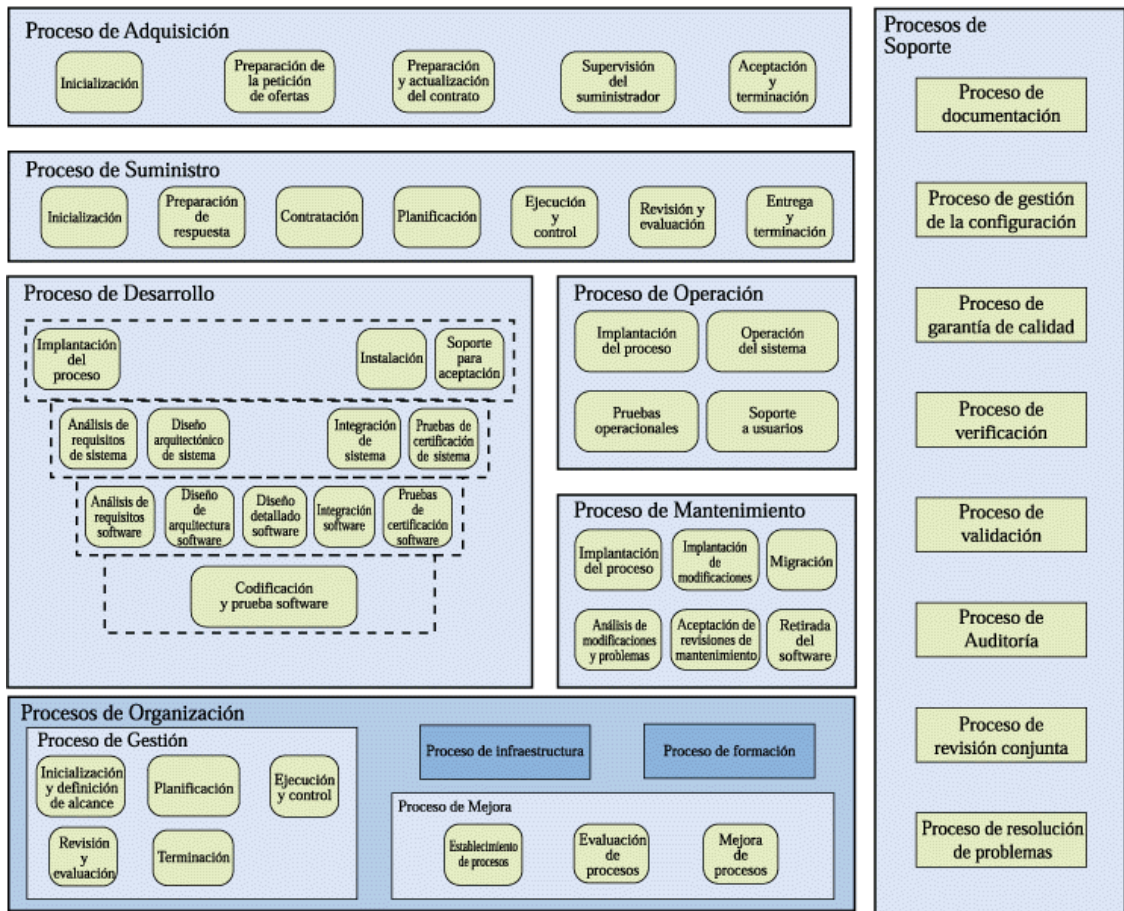


Figura 2.1

En el desarrollo de esta tesis nos enfocamos en el **Proceso de Desarrollo**, el mismo que contiene las actividades de análisis de requisitos, diseño, codificación, integración, pruebas, instalación y aceptación.

## **2.4. MODELOS Y METODOLOGIAS PARA EL PROCESO DE DESARROLLO DEL SOFTWARE**

### **2.4.1. Fundamentos**

Para resolver los problemas reales de una industria, un ingeniero del software o un equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe al proceso de desarrollo de software. Esta estrategia a menudo se llama modelo de proceso o paradigma de Ingeniería de Software. Se selecciona un modelo de proceso para la Ingeniería de Software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, los controles y las entregas que se requieran. [2]

### **2.4.2. Modelo Cascada**

- Es el primer modelo de proceso de desarrollo de software que se publicó y que se derivó de otros procesos de ingeniería. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo: 1) Análisis y definición de requerimientos. 2) Diseño de sistemas y de software. 3) Implementación y prueba de

unidades 4) Integración y prueba del sistema 5) Operación y mantenimiento. [1]

#### **2.4.3. Desarrollo evolutivo**

- Este se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. [1] La ventaja de un enfoque evolutivo es que la especificación se puede desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema.

#### **2.4.4. Desarrollo formal de sistemas**

En este enfoque, las especificaciones de requerimientos de software se refinan en una especificación formal detallada que se expresa en notación matemática. Los procesos de desarrollo de diseño, implementación y pruebas de unidades se reemplazan con un proceso de desarrollo por transformaciones donde la especificación formal se refina, a través de una serie de transformaciones, hasta llegar a un programa. [2]

#### **2.4.5. Desarrollo basado en la reutilización**

En la mayoría de los proyectos de software existe algo de reutilización de software. Por lo general, esto pasa cuando las personas que trabajan en el proyecto conocen diseños o código similares al requerido. Esta reutilización informal es independiente del proceso genérico que se utilice. Sin embargo, en años recientes, ha surgido un enfoque de desarrollo de software (Ingeniería de Software basada en componentes) que se basa en la reutilización, el cual se está volviendo muy común. [1]

#### **2.4.6. Iteración de procesos**

Para sistemas muy grandes, existe la necesidad de utilizar diferentes enfoques para las distintas partes del sistema. Los sistemas grandes están conformados usualmente de varios subsistemas para los cuales no es necesario utilizar el mismo modelo de proceso. La esencia de los procesos iterativos es que la especificación se desarrolla junto con el Software. [1]

##### **2.4.6.1. Desarrollo incremental**

El modelo de desarrollo incremental requiere que los clientes de un sistema cumplan un conjunto de requerimientos antes de que se inicie el diseño, y que el diseñador cumpla estrategias

particulares de diseño antes de la implementación. Los cambios de requerimientos durante el desarrollo implican rehacer el trabajo de captura, diseño e implementación. [2]

El proceso de desarrollo incremental tiene varias ventajas: [1]

- Los clientes no tienen que esperar hasta que el sistema completo se entregue para sacar provecho de él.
- Los clientes pueden utilizar los incrementos iniciales como un prototipo para obtener experiencia sobre los requerimientos de los incrementos para su uso inmediato.
- Existe un bajo riesgo de fallar en el proyecto total.
- Puesto que los servicios de alta prioridad se entregan primero y los incrementos posteriores se integran a ellos, entonces el primer incremento satisface los requerimientos más críticos de tal forma que el software está disponible para su uso inmediato.

#### **2.4.6.2. Desarrollo en espiral**

El modelo espiral mejora el modelo de cascada enfatizando la naturaleza iterativa del proceso de diseño. Eso introduce un ciclo de prototipo iterativo. En cada iteración, las nuevas

expresiones que son obtenidas transformando otras dadas son examinadas para ver si representan progresos hacia el objetivo. Las fases del modelo espiral son: Planteamiento de Objetivos, Identificación y reducción de riesgos, Desarrollo y Validación, y Planeación. [1]

Entre las ventajas del modelo se puede decir que:

- Centra su atención en la reutilización de componentes y eliminación de errores en información descubierta en fases iniciales.
- Los objetivos de calidad son el primer objetivo.
- Integra desarrollo con mantenimiento.
- Provee un marco de desarrollo de hardware/software.

Como desventajas se conoce que:

- Especifica el modelo del proceso y los resultados a entregar por adelantado.
- Requiere de experiencia en la identificación de riesgos.
- Requiere refinamiento para uso generalizado. [1]

#### **2.4.7. Microsoft Solution Framework**

El Marco de Trabajo de Soluciones de Microsoft (MSF) proporciona asesoramiento en lo que respecta a las fases de



diseño, creación y distribución del ciclo de vida del proyecto. Esta orientación se transmite a través de notas del producto, guías de distribución, soluciones aceleradas, kits de soluciones, estudios de casos y programas didácticos en las áreas de arquitectura de la empresa, desarrollo de aplicaciones, diseño de componentes y distribución de infraestructuras. Reconoce que la tecnología no es la única pieza de una solución exitosa. Se compone de 3 modelos: **[16]** 1) Manejo de riesgos. 2) Modelo de Equipo., y 3) Modelo de procesos. Entre las ventajas del MSF se citan: **[16]**

- Crear soluciones que unan de mejor forma los requerimientos del usuario
- Mejora exitosamente los eventos planeados.
- Mejora las competencias IT
- Maneja los riesgos del proyecto

#### **2.4.8. Rational Unified Process**

Es un proceso de desarrollo iterativo, un marco de trabajo adaptable al proceso. Las organizaciones de desarrollo y los equipos de proyecto de software seleccionan los elementos del proceso que es apropiado para sus necesidades. **[17]**

Se basa en los mejores principios y prácticas de desarrollo de software: Desarrollo iterativo, Manejo de requerimientos, Uso de

arquitectura basado en componentes, Modelos de software visuales, Verifica la calidad del software y Controles de cambio del software. Divide un proyecto en cuatro fases: Fase de Inicio, Fase de Elaboración, Fase de Construcción, y Fase de Transición. **[17]** Entre las ventajas se citan: **[18]**

- El riesgo es manejado tempranamente
- El proyecto esta orientado en brindar valor agregado a los clientes
- La arquitectura ejecutable es establecida tempranamente en el proyecto
- Todos los que contribuyen al proyecto trabajan en conjunto en un equipo
- El equipo de trabajo esta centrado en la calidad durante todo el proyecto
- Abarca todo el proceso y características necesarias para un nivel de capacidad 3 en CMM **[19]**.

Entre las desventajas se puede mencionar que si los usuarios de RUP no entienden que RUP es un marco de trabajo del proceso, puede ser percibido como pesado y costoso. **[16]**

### **2.4.9. El proceso de desarrollo de software**

Un proceso de desarrollo de software podría contener muchos subprocesos y flujos de datos, cada uno produciendo, transformando o transmitiendo productos y derivados. [7]

#### **2.4.9.1. Definición**

El proceso del software establece un marco común de actividades coherentes y resultados asociados para la producción de software que son aplicables a todos los proyectos de software, con independencia de su tamaño o complejidad. No existe un proceso ideal y diferentes organizaciones han desarrollado enfoques completamente distintos para el desarrollo de software. Los modelos y metodologías permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y a los requisitos del equipo del proyecto. [1]

#### **2.4.9.2. Características principales**

El proceso de desarrollo del software puede ser dividido en cuatro fases genéricas, con independencia del área de aplicación, tamaño o complejidad del proyecto.

Aunque existen muchos procesos diferentes de software, tienen actividades fundamentales que son comunes para todos ellos. **[1]** Estas son:

- Especificación del software.
- Diseño e implementación del software
- Validación del software
- Evolución del software

Estas actividades son consideradas como entidades dentro de la organización y del proceso de desarrollo. Las empresas podrían caracterizar estas entidades de una manera consistente y bien definida. Los atributos son características o propiedades que pueden describir las entidades, así el proceso de desarrollo puede ser descrito por atributos como el tamaño, costo, tiempo de retraso, esfuerzo realizado, productividad, progreso, inspecciones, etc. **[2]**

#### **2.4.9.3. Proyectos de software**

Es una colección de tareas de trabajo con un límite de tiempo y un producto de trabajo que debe ser entregado. El producto de trabajo entregado es software y su documentación. Los tipos de proyectos son:

- Nuevo desarrollo: Se requiere hacer un análisis y ciclo de desarrollo completo. Provee una funcionalidad nueva que no había sido desarrollada anteriormente.
- Mejora: Cambios que se realizan a un proyecto existente, donde nuevas funcionalidades pueden ser añadidas, o módulos existentes son modificados o eliminados.
- Re-desarrollo: Implica volver a desarrollar un proyecto existente que involucra un cambio en tecnología hardware o software. La funcionalidad es conocida y no requiere cambios. **[20]**

La estructura de un proyecto típicamente debe contener: redacción de la propuesta, planeación del proyecto, definición del calendario del proyecto, análisis de riesgo, selección y evaluación de recursos, costeo del proyecto, supervisión y revisión del proyecto, redacción y presentación de informes. **[1]**

## **2.5. MEDICION DEL SOFTWARE DIRIGIDO A OBJETIVOS**

El énfasis a lo largo de la medición dirigida a objetivos (Goal Question Measurement) es recolectar de información que ayude a lograr las metas de la organización, manteniendo la relación de las metas comerciales con las métricas, para que los esfuerzos de la medición tengan objetivos bien definidos. [7] La estructura del proceso de medición dirigido a objetivos utiliza las ideas y experiencias de Víctor Basili y Dieter Rombach [10]

### **2.5.1. Fundamentos**

El proceso de la medición dirigido a objetivos se basa en 3 preceptos, y 10 pasos. Los tres preceptos son:

1. La medición de objetivos se deriva de los objetivos del negocio de la empresa.
2. Concentrarse en modelos mentales proporciona contexto y enfoque.
3. GQ(I)M, GQM con Indicadores, traduce los objetivos informales en estructuras de medición ejecutables. [7]

Los 10 pasos de la medición son:
1) Identificar los objetivos de negocio
2) Identificar que necesita conocer o descubrir
3) Identificar sub-objetivos
4) Identificar entidades y atributos
5) Formalizar objetivos de medición
6) Identificar preguntas de medición e indicadores
7) Identificar los elementos a medir
8) Definir y documentar las mediciones e indicadores
9) Identificar acciones para implementar las mediciones
10) Preparar un plan.

**Tabla 2.1 Medición de software basado en objetivos**

Es importante tener presente que todo proceso opera en un contexto que contribuye o disminuye las perspectivas para el éxito. La información cuantitativa sobre los factores del contexto ayudará a interpretar los datos que se recogen de las entidades examinadas. [7]

### **2.5.2. Aplicación del proceso de medición de software dirigido a objetivos**

El proceso dirigido a objetivos empieza con identificar las metas comerciales y descomponerlos en sub-objetivos manejables. Finaliza con un plan para llevar a cabo las medidas definidas e indicadores que sustentan las metas. A lo largo del proceso, mantiene “trazabilidad” o la relación hacia atrás a las metas, para

que al recolectar y procesar los datos de la medición no se pierdan de vista los objetivos. [7] A continuación se resumen los pasos que se deben seguir en el proceso de medición dirigido a objetivos tal y como se indican en el capítulo 4 del libro “Goal-driven software measurement - A guidebook” [7].

#### **2.5.2.1. Identificación de los objetivos del negocio**

El primer paso es identificar las metas del negocio a las que la organización dirige sus esfuerzos, las metas en organizaciones jerárquicas están relacionadas y vienen desde abajo compitiendo en base a la productividad, flexibilidad, satisfacción del cliente, problemas de calidad, entrega, ciclo de tiempo, y pérdidas. [7]

#### **2.5.2.2. Identificar que se busca conocer o aprender**

Con las metas del negocio identificadas, el próximo paso es identificar “cosas” que se deberían saber para entender, evaluar, predecir, controlar, mejorar, o motivar elementos de su organización con el objetivo de lograr las metas identificadas. [7]

El proceso de medición crea modelos mentales consistentes de los procesos relevantes que se pretende medir guiándose



por lo que se quiere mejorar y los problemas que se deben enfrentar para hacerlo, luego lista las entidades importantes en el proceso a medir, asegurándose de que cubran cada una de las 4 clases de entidades de procesos: entradas y recursos, productos y derivados de productos, artefactos internos y trabajo dentro del proceso, actividades y flujos de tareas. Después para cada entidad, se listan preguntas que pueden ayudar a planear y dirigir el progreso hacia las metas. [7]

#### **2.5.2.3. Identificar Sub-objetivos**

El tercer paso en el proceso dirigido a objetivos es traducir las metas u objetivos de alto-nivel en sub-objetivos que se relacionan específicamente por las actividades que se realizan. Las preguntas que fueron sugeridas por el modelo mental apuntan, implícita o explícitamente, a las entidades y atributos asociados con alcanzar las metas. [7]

#### **2.5.2.4. Identificar entidades y atributos**

El próximo paso es usar las preguntas para refinar el modelo mental del proceso operacional, las entidades y los atributos asociados al proceso. [7]

Se listan entonces los atributos pertinentes asociados con cada entidad, que al cuantificarlos ayudarán a contestar las preguntas o establecer un contexto para interpretar las respuestas. El punto simplemente es identificar características importantes de la entidad y el contexto que ayudará a dar visión a las preguntas, se debe releer cada lista conscientemente, buscando entradas que parecen implicar medidas únicas. [7]

#### **2.5.2.5. Formalizar los objetivos de medición**

En los siguientes pasos se complementa el paradigma agregando un "indicador" al paso entre el Q y M de **(GQM)** normal. Primero, se debe establecer un fundamento sólido para GQ(I)M identificando objetivos de medición y preparando declaraciones estructuradas para estos.

Por "indicador," se quiere decir, un cuadro o reporte que nos ayude a responder la pregunta de la mejor forma. El aspecto estructurado es importante, porque asegura que los puntos claves no se pasan por alto cuando se centren en definición y recolección de medidas. [7]

Los objetivos de medición bien-estructurados tienen cuatro componentes: Objeto de interés (una entidad), Propósito, Perspectiva, y Descripción del contexto y limitaciones [7].

#### **2.5.2.6. Identificar preguntas cuantificables e indicadores**

Los objetivos de medición estructurados construidos en el punto anterior proveen un fundamento sólido para los pasos del paradigma (**GQM**). En este punto los objetivos del negocio han sido descompuestos y refinados en entidades significativas, propósitos, perspectivas, y contextos que pueden ser identificados. El "indicador" (**I**) sirve para comunicar o explicar la importancia de los resultados de la medición que se diseña. Esto permite distinguir claramente entre el concepto de comunicación y de medición. [1]

#### **2.5.2.7. Identificar los datos a recopilar**

Con los indicadores que se requieren trazar o graficar, se pasa a identificar los elementos de datos que se deben recopilar para crear los indicadores [7].

En este paso y en el próximo se debe identificar los elementos de los datos y definir cómo se recolectarán las medidas. Para completar la primera tarea, simplemente se

debe hacer una lista de todos los elementos de datos que se necesitan recolectar [7].

#### **2.5.2.8. Definir las mediciones**

El punto clave del éxito de la metodología es poder decir exactamente cómo se obtiene cada medida, para que se puedan interpretar los valores correctamente cuando se usen los datos que fueron reunidos o informados por alguien más [7]. El problema primario es hacer comprender completamente lo que representan los valores medidos. Cuando los usuarios de los datos no conocen cómo fueron recopilados, hacen asunciones inválidas. Esto lleva a interpretaciones incorrectas, análisis impropios, y las decisiones erróneas. Se deben satisfacer dos criterios importantes: Comunicación y Repetibilidad. [7]

#### **2.5.2.9. Identificar acciones necesarias para implementar las mediciones**

En este punto se debe congregar la información sobre el estado actual y uso de mediciones, para preparar un plan efectivo de implementación de las mediciones que se han definido. Se siguen las siguientes actividades: [7]

Se debe realizar: **Análisis**, identificando mediciones que la organización usa actualmente y entender cómo son recopiladas. **Diagnóstico**, determinando la conveniencia y disponibilidad de datos existentes e identificar fuentes potenciales y actuales para la información. **Acción**, traduciendo los resultados de su análisis y diagnósticos en pasos a implementar. [7]

#### **2.5.2.10. Preparar el plan de medición**

Una vez que se conoce con que comenzar (análisis), que tan bien resuelven sus necesidades del negocio las medidas actuales (diagnosís), y las acciones para conocer las necesidades restantes (acción), se puede preparar un plan para ejecutar las acciones de medición. Una **Plantilla de Medición** sirve para ayudar a identificar y a estructurar el plan y se compone de: Objetivo, Descripción (Background, Objetivos, Alcance, Relación con Otros Esfuerzos para el Mejoramiento del Proceso del Software, y Relación con Otras Actividades Funcionales), Implementación (Actividades, Productos, y Tareas, Cronograma, Recursos, Responsabilidades, Medición y Monitoreo, Asunciones, y Administración de Riesgo), y Operación Sostenida. [7]

## **2.6. HERRAMIENTAS ESCOGIDAS PARA LA METODOLOGIA GQIM**

El análisis de los datos medidos permite obtener información sobre la calidad del producto, estudiar y corregir el funcionamiento del proceso, y tomar decisiones basándose en el análisis de los datos. Al momento de realizar este análisis es necesario recurrir a técnicas o herramientas estadísticas que permitan resumir y visualizar estos datos. A continuación se describen algunas técnicas y herramientas que se pueden utilizar para los propósitos antes mencionados.

### **2.6.1. Técnica nominal de grupos**

Aprovecha los juicios de una variedad de personas con conocimiento y habilidades diferentes para generar, registrar, discutir y dar prioridad a un conjunto de ideas sobre un tópico específico. **[15]**

Los beneficios que provee esta técnica son: la participación equilibrada entre los miembros, produce ideas más creativas, produce un número mayor de ideas que los grupos relacionados por áreas tradicionales, los resultados satisfacen más a los participantes, incrementa las reuniones cara a cara en grupos, incentiva a que los participantes confronten diferencias en base a

resolver problemas, deja un mejor sentido de cierre y cumplimiento. **[15]**

Entre las limitaciones que afronta se puede decir que requiere de preparación de antemano lo que significa que no puede ser una técnica espontánea, se limita a un solo propósito, es difícil de cambiar de tema en medio de la reunión, se requiere el acuerdo de todos los participantes para usar el mismo método estructurado en otros temas. **[15]**

### **2.6.2. Valor devengado**

Es una técnica de administración de proyectos que rastrea objetivamente el trabajo cumplido. Se diferencia de otras por combinar medidas de desempeño técnico, cumplimiento del trabajo planeado, cumplimiento del horario, y comportamiento del costo dentro de una sola metodología integrada. Proporciona una advertencia de problemas del desarrollo mientras hay tiempo para corregirlos, mejora la definición de alcance del proyecto, comunica el progreso a los interesados, y mantiene al equipo del proyecto enfocado en lograr progreso. **[21]**

Entre las limitaciones se puede decir que si la aplicación de valor devengado no se descompone puede ser demasiado costosa. No

mide la calidad del proyecto, sirve para indicar si un proyecto está bajo el presupuesto, a tiempo con el horario y el alcance ejecutado, pero puede tener clientes insatisfechos y los resultados finalmente infructuosos. Su uso asume que los interesados se preocupan de medir el progreso objetivamente.

**[21]**

### **2.6.3. Estadística descriptiva**

Formula reglas y procedimientos para la presentación de una masa de datos en una forma más útil y significativa. Establece normas para la representación gráfica de los datos. También son una base importante para el análisis en casi todas las disciplinas académicas. Se dedica al ordenamiento y tratamiento mecánico de la información para su presentación por medio de tablas y de representaciones gráficas, así como de la obtención de algunos parámetros útiles para la explicación de la información. **[22]**

### **2.6.4. Histogramas**

Un histograma es un gráfico o diagrama que muestra el número de veces que se repiten cada uno de los resultados cuando se realizan mediciones sucesivas. Esto permite ver alrededor de que valor se agrupan las mediciones (Tendencia Central) y cual es la



dispersión alrededor de ese valor central. [23] Se agrupan los datos en intervalos contando los resultados de las mediciones dentro de cada intervalo (frecuencia), después se representan las frecuencias en gráficos bi-dimensionales, con columnas de altura proporcional a la frecuencia de cada intervalo. Permite visualizar rápidamente información que estaba oculta en la tabla original de datos. Nos permite apreciar la Tendencia Central de las mediciones. Además de observar la Dispersión de las mediciones. [24]

#### **2.6.5. Gráficos de Pareto**

El Diagrama de Pareto es un histograma especial, en el cual las frecuencias de ciertos eventos aparecen ordenadas de mayor a menor [24]. Mediante los diagramas de Pareto se pueden detectar los problemas que tienen más relevancia, los elementos más importantes se ubican listando todos los elementos por orden descendente. La gráfica es muy útil al permitir identificar visualmente en una sola revisión tales minorías de características vitales a las que es importante prestar atención. [9]

### **2.6.6. Gráficos de dispersión**

Los Diagramas de Dispersión o Gráficos de Correlación permiten estudiar la relación entre 2 variables. Dadas 2 variables X e Y, se dice que existe una correlación entre ambas si cada vez que aumenta el valor de X aumenta proporcionalmente el valor de Y (Correlación positiva) o si cada vez que disminuye el valor de X disminuye en igual proporción el valor de Y (Correlación negativa). En un gráfico de correlación se representa cada par de variables o características X, Y como un punto donde se cortan las coordenadas de X e Y. [24]

### **2.6.7. Diagrama causa-efecto**

El valor de una característica de calidad depende de una combinación de variables y factores que condicionan el proceso. La variabilidad de las características de calidad es un efecto observado que tiene múltiples causas. Cuando ocurre algún problema con la calidad del producto, se debe investigar para identificar las causas del mismo. Para esto se utilizan los Diagramas de Causa-Efecto, conocidos también como Diagramas de Espina de Pescado por la forma que tienen. Estos diagramas fueron utilizados por primera vez por Kaoru Ishikawa [23]. Un diagrama de Causa-Efecto es educativo, sirve para que la gente

conozca en profundidad el proceso con el que trabaja, visualizando con claridad las relaciones entre los Efectos y sus Causas. Sirve también para guiar las discusiones, al exponer con claridad los orígenes de un problema de calidad. Y permite encontrar rápidamente las causas asignables cuando el proceso se aparta de su funcionamiento habitual. [23]

#### 2.6.8. Complejidad ciclométrica

Es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa [2]. Está enfocada a medir atributos del software, de forma opuesta a otras métricas como los reportes de fallas y/o avances de proyecto. Una medición ideal de complejidad debe contener dos componentes:

- **Descriptivo** que identifica código susceptible a errores, difícil de entender, difícil de modificar, difícil de probar y demás.
- **Prescriptivo** que identifica los pasos operacionales para ayudar a controlar el software, por ejemplo: indicar la cantidad de pruebas que deben ejecutarse para cada módulo.

La métrica desarrollada por Tomas McCabe para medición de complejidad mide el número de decisiones lógicas en un segmento

de código y cumple con los dos componentes antes mencionados.

**[25]**

Existe una fuerte conexión entre la complejidad ciclométrica y las pruebas de software, la complejidad es una fuente común de errores en el software, consecuentemente, la complejidad puede ser utilizada para definir el esfuerzo de pruebas, esto se logra haciendo énfasis en aquellos elementos de software de alta complejidad, que tienen mayor susceptibilidad a error. En Complejidad Ciclométrica esto se define como “Test Paths” o el mínimo número de casos de prueba necesarios para satisfacer criterios. **[25]**

#### **2.6.9. Encuestas**

La encuesta es un método de obtener información de una muestra de individuos que es usualmente sólo una fracción de la población bajo estudio. Tienen una gran variedad de propósitos, y pueden conducirse de muchas maneras, incluyendo por teléfono, por correo o en persona. **[26]**

Todas las encuestas tienen algunas características en común: **[26]**

- A diferencia de un censo, las encuestas recogen información de una porción de la población de interés, dependiendo el tamaño de la muestra en el propósito del estudio.
- La información es recogida usando procedimientos estandarizados de manera que a cada individuo se le hacen las mismas preguntas y de manera similar.
- La intención de la encuesta no es describir los individuos particulares quienes, por azar, son parte de la muestra sino obtener un perfil compuesto de la población.
- El estándar de la industria para todas las organizaciones respetables que hacen encuestas es que los participantes individuales nunca puedan ser identificados al reportar los hallazgos. Todos los resultados de la encuesta deben presentarse en resúmenes completamente anónimos, tal como tablas y graficas estadísticas.

CAPITULO

3

Procedimiento

# CAPITULO 3

## PROCEDIMIENTO

### 3.1. DISEÑO DE LA INVESTIGACIÓN

A continuación se explica el tipo de investigación escogido para la tesis, cómo se seleccionó el ámbito de estudio para enfocarse en el desarrollo de software y las fases principales en las que se organizó nuestra investigación.

#### 3.1.1. Tipo de investigación

El tipo de investigación exploratorio que se describe en el capítulo 4 del libro Metodología de la investigación [19], se escogió para la tesis ya que en el contexto de nuestro trabajo de investigación se recomienda por las siguientes razones:

- 1) Las métricas y el proceso de desarrollo del software son temas que han sido escasamente estudiados en nuestro medio.
- 2) Para aumentar el grado de conocimiento con estos temas relativamente desconocidos.

### **3.1.2. Selección del ámbito de estudio**

Un significativo número de empresas que fueron estudiadas por la ESPOL [4], antes de la realización de esta tesis, estaban ubicadas en las ciudades de Quito (47) y Guayaquil (13), en consecuencia, las empresas ubicadas en estas ciudades serían el objeto de estudio de nuestra tesis.

Por otra parte, el proceso desarrollo de software no es el único proceso productivo que se ejecuta en una empresa de software. Sin embargo para limitar el ámbito de estudio de la tesis, este es el proceso de nuestro interés.

### **3.1.3. Fases principales de la investigación**

Para comenzar es necesario tener alguna referencia que permita tener un punto de partida con el cual trabajar. Existía información referente a la ubicación, cantidad, herramientas utilizadas, personal y tamaño de las empresas, entre otros datos de utilidad principalmente obtenidos en el “Estudio estadístico exploratorio de las empresas de desarrollo de software asentadas en Quito, Guayaquil y Cuenca.” [4] No obstante, no se tenía ni se conocía información de aspectos de la calidad y/o métricas referentes a software utilizadas por empresas ecuatorianas.



Por tal motivo la tesis inicio con un **estudio exploratorio** orientado a identificar empresas desarrolladoras de software y determinar el escenario de estudio mencionado. De esta forma se “prepara el terreno”, facilitando guías para la ejecución de las siguientes actividades de la tesis.

Con los datos obtenidos en la fase descrita anteriormente, fue posible aplicar la “**medición del software basado en objetivos**” que se explicó en el capítulo 2 con el propósito de lograr un conjunto de objetivos de medición, indicadores y los formularios adecuados para la recopilación de datos.

A continuación, en tres empresas de software se escogió un proyecto de desarrollo de software disponible para la **aplicación de estas herramientas** y finalmente **analizar los datos** obtenidos de estos proyectos.

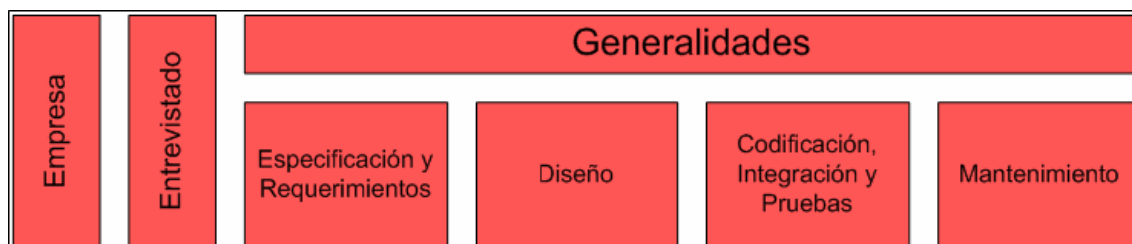
## **3.2. ESTUDIO EXPLORATORIO**

### **3.2.1. Formación del instrumento**

Utilizando información de cómo construir instrumentos de medición descritos en el capítulo 9 del libro “Metodología de la Investigación” [19] y datos de trabajos de investigación similares realizados en otros países [27], se construyó una primera versión de un cuestionario dirigido a los gerentes de las empresas de software ecuatorianas. Esta versión pasó por la revisión de profesionales y docentes nacionales y extranjeros con el propósito de limitar el contenido y mejorar la redacción de las preguntas. Posteriormente se aplicó a 15 empresas desarrolladoras de software de la ciudad de Guayaquil como prueba piloto, obteniéndose la versión final que se aplicaría al resto de empresas de Guayaquil y Quito.

El cuestionario comprende siete categorías: Empresa, Entrevistado, Generalidades del desarrollo de software, Especificación y requerimientos, Diseño, Codificación, Integración y pruebas, y Mantenimiento. La figura 3.1 muestra el esquema del cuestionario.

### Esquema del cuestionario exploratorio



**Figura 3.1.**

A continuación se listan las variables utilizadas en el estudio exploratorio clasificadas por su nivel de medición que puede ser nominal, ordinal y de razón:

- **Nivel de medición nominal**

**Dicotómicas:** Género, escolaridad, certificación ISO 9001:2000, exportación de software, conocimiento de métrica de software, tipos de métricas, medición del proceso de desarrollo, registro de horas por actividad, principal fuente de defectos, re-utilización de componentes.

**Catóricas:** Cargo, documentación del desarrollo de software, dificultad para documentar, modelo del proceso de software, mayor consumo de recursos, tipos de métricas, tamaño del software, medición de factores del desarrollo, actividades registradas por horas, toma de decisiones usando mediciones, recopilación de datos para mediciones, estimación en la especificación y requerimientos, uso de

datos históricos, herramientas de planificación, inspecciones en el diseño, porcentaje de defectos detectados en diseño, tipo de diseño, estimación después del diseño, porcentaje de componentes re-utilizados, tiempo y recursos consumidos re-utilizando software, control de defectos, personal de pruebas, tipo de pruebas, pruebas de cobertura de código, porcentaje de defectos detectados en pruebas, contingencia frente a muchos defectos, corrección de errores reportados, registro del tiempo en hacer correcciones, pérdida de datos por errores, realización de cambios pedidos por el cliente, registro del tiempo en hacer cambios.

- **Nivel de medición ordinal**

Éstas incluyen: metas de la empresa al desarrollar productos de software, problemas en especificación y requerimientos, factores que aumentan el costo de mantenimiento, tipo de mantenimiento.

- **Nivel de medición de razón**

Estas variables contemplan: edad, años de trabajo, empleados, años de la empresa, valor facturado, software desarrollado, personal asignado a proyectos, tiempo de desarrollo, costo de desarrollo, distribución de los recursos, defectos remanentes en el software, porcentaje de reinstalaciones.

### **3.2.2. Validación y aplicación del instrumento**

El construir este cuestionario teniendo en cuenta trabajos similares, la revisión por parte de profesionales y docentes del área de software y la prueba piloto, ayudan a obtener la característica de validez en los contenidos del cuestionario. Además, para verificar la confiabilidad del instrumento y considerando que el cuestionario consta de indicadores de tipo dicotómico, numérico y nominal múltiple se tomó en cuenta el coeficiente Alfa de Cronbach estandarizado determinándose un valor de 0.838 que demuestra un nivel aceptable de consistencia interna entre las variables del instrumento.

La aplicación de este cuestionario a 61 empresas de software ubicadas en Quito y Guayaquil se dió entre los meses de Abril y Mayo del año 2005.

El resumen de los datos recopilados, y las conclusiones obtenidas son una descripción actual del las empresas de software ecuatorianas y por ende el escenario de estudio o punto de partida que nos interesaba establecer.

### **3.3. DESARROLLO DEL GQIM**

Teniendo en cuenta el marco teórico descrito en el capítulo 2, por cada actividad que comprende el proceso GQIM, en las siguientes secciones se presentan los entregables que se generaron en cada uno de los pasos de este proceso. Sin embargo, a partir del paso 5 (Objetivos de medición), solo se presenta una parte de esos resultados, con el propósito de evitar repetir la información y extender el tamaño de este documento pues los resultados completos se pueden observar en el apéndice “Plan de métricas”.

#### **3.3.1. Objetivos prioritarios de negocio de las empresas de software**

Ya que uno de los objetivos de la investigación es elaborar un plan de métricas que sea lo suficientemente general para que pueda ser adaptado a diferentes empresas de software, es necesario empezar partiendo de objetivos comunes e importantes para la mayoría de las empresas de software. Por esto dentro del estudio exploratorio se aplicó una adaptación de la “técnica nominal de grupos” que nos permitió determinar la importancia y prioridad de un conjunto de objetivos de negocio. De éstos seleccionamos los tres primeros.

Ref.	Objetivos de negocio	Prioridad
O1	Maximizar la <b>satisfacción del cliente</b>	1ero con el 98%
O2	Mejorar la <b>calidad</b> del producto de software	2do con el 84%
O3	Mejorar la <b>productividad</b>	3ero con el 84%

**Tabla 3.1. Objetivos del Negocio - Ref**

En los siguientes pasos de GQIM los objetivos escogidos se identificarán con los códigos de referencia listados en la columna titulada “**Ref.**”

### 3.3.2. Lista entidad-pregunta

Teniendo presente los objetivos de negocio identificados en el paso anterior y desde el punto de vista del administrador de proyectos, se identificaron las entidades de mayor interés del proceso de desarrollo de software y por cada entidad identificada se generaron preguntas orientadas a lograr los objetivos de negocio listados en el paso anterior.

Ref.	Ref.	Entradas y Recursos	Preguntas relacionadas a los objetivos de negocio
O1	1	Cliente	¿Están nuestros clientes satisfechos con el SW que producimos?
O2	2		¿Cuales son los factores del SW que afectan la satisfacción del cliente?
O3	3		¿En que fases del desarrollo del SW está involucrado

			el cliente?
--	--	--	-------------

**Tabla 3.2 Entradas y recursos.**

Ref.	Ref.	Entradas y Recursos	Preguntas relacionadas a los objetivos de negocio
O1	4	Personal	¿La experiencia del personal asignado al proyecto favorece el desarrollo del SW?
O2	5		¿En que medida, la rotación del personal afecta el desarrollo del SW?
O3	6		¿Se asigna al proyecto de SW el personal necesario para el desarrollo del mismo?
O1	7	Consumibles	¿Se han cambiado el tiempo y costo establecidos durante la realización del proyecto?
O3	8		¿Las estimaciones de tiempo y costo son apropiadas para desarrollar el proyecto?
O1	9	Herramientas	¿Que herramientas se prefiere utilizar durante el desarrollo del proyecto?
O2	10		¿Los diferentes entregables principalmente los diseños se desarrollaron utilizando herramientas CASE?
O3	11		¿Las herramientas utilizadas permiten mejorar el desempeño del personal?

**Tabla 3.2 Entradas y recursos (continuación).**

Ref.	Ref.	Actividades y flujos	Preguntas relacionadas a los objetivos de negocio
O1	12	Planeación del proyecto	¿El cliente participó en el desarrollo de los proyectos?
O2	13		¿El proyecto se planifica usando datos de referencia de proyectos anteriormente realizados?
O3	14		¿El proyecto se desarrolla de acuerdo al cronograma establecido?
O1	15	Especificación de requerimientos	¿Los requerimientos están adecuadamente definidos al inicio del proyecto?
O2	16		¿Se utiliza el tiempo necesario para el buen establecimiento de los requerimientos?
O3	17		¿Se cuenta con el apoyo de personas con



		conocimiento del dominio del negocio?
--	--	---------------------------------------

**Tabla 3.3 Actividades y flujos**

Ref.	Ref.	Actividades y flujos	Preguntas relacionadas a los objetivos de negocio
O1	18	Diseñar	¿Los diseños son revisados y aprobados por el cliente?
O2	19		¿Que proporción del tiempo del proyecto se dedica al diseño?
O3	20		¿Se toman en cuenta patrones de diseño en el desarrollo del proyecto?
O1	21	Construir	¿El SW se construye utilizando algún estándar de codificación?
O2	22		¿En que proporción se han reutilizado componentes de software en nuevos proyectos?
O3	23		¿Cuanto tiempo toma construir el SW?
O1	24	Probar	¿Cuanto tiempo se emplea en probar el software desarrollado?
O2	25		¿Se ejecutaron las pruebas suficientes en el SW?
O3	26		¿Qué efecto tienen en la duración del proyecto los defectos detectados en las pruebas?
O1	27	Instalar	¿Se presentan defectos durante la instalación y posterior uso del SW?
O2	28		¿La decisión de liberar el producto está debidamente planificada?
O3	29		¿La instalación del SW se hace en el tiempo planificado?
O1	30	Inspecciones y Reuniones	¿Con qué frecuencia el cliente participa en reuniones?
O2	31		¿Con qué frecuencia se realizan inspecciones?
O3	32		¿Cuánto tiempo emplea el personal para realizar reuniones?

**Tabla 3.3 Actividades y flujos (continuación)**

Ref.	Ref.	Artefactos internos	Preguntas relacionadas a los objetivos de negocio
O2	33	Presupuesto	¿Se cuenta con el presupuesto adecuado para todas las actividades planificadas?
O3	34		¿Se dan cambios al presupuesto durante el desarrollo del proyecto?
O2	35	Cronograma	¿El tiempo establecido es el adecuado para el desarrollo de SW de calidad?
O3	36		¿Se conoce la diferencia entre el tiempo planificado y el consumido durante el proyecto?

**Tabla 3.4. Artefactos internos.**

Ref.	Ref.	Productos y derivados	Preguntas relacionadas a los objetivos de negocio
O1	37	Requerimientos	¿Cuántos requerimientos están desarrollados y cuántos faltan desarrollar?
O2	38		¿Están los requerimientos clara y completamente definidos?
O3	39		¿De que forma cambian los requerimientos durante el desarrollo del proyecto?
O1	40	Diseño del SW	¿El diseño del SW, está aprobado por el cliente y/o usuario final?
O2	41		¿Los diseños están libres de defectos?
O3	42		¿Se realizan inspecciones sobre los diseños (Quién, con qué frecuencia)?
O1	43	Código fuente	¿Se utiliza algún estándar para la codificación?
O2	44		¿El código fuente esta libre de defectos cuando es liberado?
O3	45		¿Se han re-utilizado componentes de SW?
O1	46	Pruebas	¿El cliente ayuda a establecer los escenarios que servirán para probar el SW?
O2	47		¿Cuánto tiempo se dedica para las pruebas del SW?
O3	48		¿Cuál es la cantidad de defectos descubiertos y cuánto toma corregirlos?

**Tabla 3.5 Productos y derivados.**

Ref.	Ref.	Productos y derivados	Preguntas relacionadas a los objetivos de negocio
O1	49	Documentación	¿Se dispone de una adecuada documentación sobre el desarrollo del proyecto?
O2	50		¿Se utilizaron plantillas para la documentación del proyecto?
O3	51		¿Cuál es el esfuerzo necesario para llevar la documentación del desarrollo del SW?
O1	52	Conocimientos y Experiencias	¿El personal cuenta con experiencia en el dominio de negocio del cliente?
O2	53		¿Es necesario capacitar al personal durante el desarrollo del proyecto?
O3	54		¿Los involucrados en el desarrollo del proyecto cuentan con el apoyo de expertos?

**Tabla 3.5 Productos y derivados (continuación).****3.3.3. Sub-Objetivos de negocio**

En el paso anterior, lo importante era generar un listado de entidades y preguntas, sin poner mayor cuidado en la redacción o ubicación de la pregunta adecuada con la entidad adecuada. De este aspecto nos preocupamos en este paso, en el que además por cada agrupación de entidades y sus correspondientes preguntas se elabora un sub-objetivo que hace más específicos y concretos a los objetivos de negocio que tienen la característica de ser de alto nivel.

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	1	¿Están nuestros clientes satisfechos con el SW que producimos?
O1	12	¿El cliente participó en el desarrollo de los proyectos?
O1	18	¿Los diseños son revisados y aprobados por el cliente?
O1	30	¿Con que frecuencia el cliente participa en reuniones?
O1	40	¿El diseño del SW, esta aprobado por el cliente y/o usuario final?
O1	46	¿El cliente ayuda a establecer los escenarios que servirán para probar el SW?
O2	2	¿Cuáles son los factores del SW que afectan la satisfacción del cliente?
O3	3	¿En que fases del desarrollo del SW esta involucrado el cliente?
<b>Sub. Objetivo 1</b>		<b>Conocer el nivel de satisfacción del cliente y su participación en el proyecto.</b>

**Tabla. 3.6. Agrupación Nro. 1. Cliente.**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	4	¿La experiencia del personal asignado al proyecto favorece el desarrollo del SW?
O1	52	¿El personal cuenta con experiencia en el dominio de negocio del cliente?
O2	5	¿En que medida, la rotación del personal afecta el desarrollo del proyecto de software?
O2	53	¿Es necesario capacitar personal durante el desarrollo del proyecto?
O3	6	¿Se asigna al proyecto de software el personal necesario para el desarrollo del mismo?

**Tabla. 3.7. Agrupación Nro. 2: Personal, conocimientos y experiencias.**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O3	17	¿Se cuenta con el apoyo de personas con conocimiento del dominio del negocio?
O3	54	¿Los involucrados en el desarrollo del proyecto cuentan con el apoyo de expertos?
Sub. Objetivo 2		<b>Mejorar la estimación del personal requerido y la planificación de las capacitaciones.</b>

**Tabla. 3.7. Agrupación Nro. 2: Personal, conocimientos y experiencias (continuación)**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	9	¿Qué herramientas se prefiere utilizar durante el desarrollo del proyecto?
O2	10	¿Los diferentes entregables principalmente los diseños se desarrollaron con herramientas CASE?
O3	11	¿Las herramientas utilizadas mejoran el desempeño del personal?
Sub. Objetivo 3		<b>Conocer la utilidad de las herramientas para el desarrollo de SW.</b>

**Tabla. 3.8. Agrupación Nro. 3: Herramientas**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	49	¿Se dispone de una adecuada documentación del proyecto?
O2	50	¿Se utilizaron plantillas para la documentación del proyecto?
O3	51	¿Cuál es el esfuerzo necesario para documentar el desarrollo del SW?
Sub. Objetivo 4		<b>Evaluar el impacto de documentar o no documentar los proyectos de software.</b>

**Tabla 3.9. Agrupación Nro. 4: Documentación**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	7	¿Se han cambiado el tiempo y costo establecidos durante la realización del proyecto?
O1	21	¿El SW se construye utilizando algún estándar de codificación?
O1	24	¿Cuanto tiempo se emplea en probar el software desarrollado?
O2	13	¿El proyecto se planifica usando datos de referencia de proyectos anteriormente realizados?
O2	16	¿Se utiliza el tiempo necesario para el buen establecimiento de los requerimientos?
O2	19	¿Que proporción del tiempo del proyecto se dedica al diseño?
O2	25	¿Se ejecutaron las pruebas suficientes en el SW?
O2	28	¿La decisión de liberar el producto esta debidamente planificada?
O2	31	¿Con que frecuencia se realizan inspecciones?
O2	33	¿Se cuenta con el presupuesto adecuado para todas las actividades planificadas?
O2	35	¿El tiempo establecido es el adecuado para el desarrollo de SW de calidad?
O2	47	¿Cuanto tiempo se dedica para las pruebas?
O3	8	¿Las estimaciones de tiempo y costo son apropiadas para desarrollar el proyecto?
O3	14	¿El proyecto se desarrolla de acuerdo al cronograma establecido?
O3	23	¿Cuánto tiempo toma construir el SW?
O3	29	¿La instalación del SW se hace en el tiempo planificado?
O3	32	¿Cuánto tiempo emplea el personal para realizar reuniones?
O3	34	¿Se dan cambios al presupuesto durante el desarrollo del proyecto?

**Tabla. 3.10. Agrupación Nro. 5.  
Proceso de desarrollo del software, (Consumibles, planeación del proyecto, cronograma, presupuesto)**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O3	36	¿Se conoce la diferencia entre el tiempo planificado y el consumido durante el proyecto?
<b>Sub. Objetivo</b>		<b>Planificar con mejor precisión el proyecto</b>

**Tabla. 3.10. Agrupación Nro. 5.  
Proceso de desarrollo del software (continuación)**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	15	¿Los requerimientos están adecuadamente definidos al inicio del proyecto?
O1	37	¿Cuántos requerimientos están desarrollados y cuántos faltan desarrollar?
O2	38	¿Están los requerimientos clara y completamente definidos?
O3	39	¿De qué forma cambian los requerimientos durante el desarrollo del proyecto?
<b>Sub. Objetivo 5</b>		<b>Planificar con mejor precisión el proyecto</b>
<b>Sub. Objetivo 6</b>		<b>Monitorear los cambios en los requerimientos durante el desarrollo del proyecto</b>

**Tabla. 3.11. Agrupación Nro. 6. Requerimientos.**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O3	20	Se toman en cuenta patrones de diseño en el desarrollo del proyecto?
O3	42	Se realizan inspecciones sobre los diseños (Quién, con qué frecuencia)?
Sub. Objetivo 5		<b>Planificar con mejor precisión el proyecto</b>
Sub. Objetivo 7		<b>Generar diseños confiables</b>

**Tabla. 3.12. Agrupación Nro. 7. Diseño.**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	43	Se utiliza algún estándar para la codificación?
O2	22	En que proporción se reutilizan componentes de software?
O3	45	Se han re-utilizado componentes de SW?
Sub. Objetivo 5		<b>Planificar con mejor precisión el proyecto</b>
Sub. Objetivo 8		<b>Disminuir el tiempo de codificación del SW</b>

**Tabla 3.13. Agrupación Nro. 8. Código fuente.**

Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O1	27	Se presentan defectos durante la instalación y posterior uso del SW?
O2	41	Los diseños están libres de defectos?
O2	44	El código fuente está libre de defectos cuando es liberado?
O3	26	Qué efecto tienen en la duración del proyecto los defectos detectados en las pruebas?

**Tabla 3.14. Agrupación Nro. 9. Defectos.**



Ref.	Ref.	Preguntas relacionadas a los objetivos de negocio
O3	48	Cuál es la cantidad de defectos descubiertos y cuánto tiempo toma corregirlos?
Sub. Objetivo 5		<b>Planificar con mejor precisión el proyecto</b>
Sub. Objetivo 9		<b>Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final</b>

**Tabla 3.14. Agrupación Nro. 9. Defectos (continuación)**

Esta última agrupación reemplaza a la entidad “Pruebas”, pues las preguntas que se identificaron para esta entidad tenían más relación con la entidad “Defectos” y además ya está tomada en cuenta de forma implícita en la agrupación Nro. 5, “Proceso de desarrollo del software”.

#### **3.3.4. Entidades y atributos**

De forma implícita o explícita, en cada sub-objetivo y preguntas relacionadas a los objetivos de negocio que se clasificaron en el paso anterior se pueden observar varias entidades que pueden ser consideradas para la formación estructurada de objetivos de medición. El propósito de este paso es determinar las entidades de nuestro interés y sus respectivos atributos.

<b>Objetivos de negocio:</b> O1, O2, O3.	
Sub. Objetivo Nro. 1.	Conocer el nivel de satisfacción del cliente y su participación en el proyecto.
Pregunta(s) Nro.	1, 2, 3.
<b>Entidad</b>	<b>CLIENTE / USUARIO FINAL</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Tipo.	Actividad comercial, financiera, industrial, etc.
2. Participación.	Del cliente y/o usuario final en el desarrollo del SW.
3. Satisfacción.	Con el software y con la gestión del proyecto de software.

**Tabla 3.15. Cliente / Usuario final**

<b>Objetivos de negocio:</b> O1, O2, O3.	
Sub. Objetivo Nro. 2.	Mejorar la estimación del personal requerido y la planificación de las capacitaciones.
Pregunta(s) Nro.	4, 52, 5, 53, 6, 54.
<b>Entidad:</b>	<b>PERSONAL</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Disponibilidad.	Una misma persona puede trabajar en varios proyectos.
2. Composición.	Máximo nivel de educación y rol del personal
3. Conocimiento.	De las herramientas necesarias para el desarrollo del SW.
4. Experiencia.	En el dominio de negocio del cliente y/o en desarrollo de proyectos.
5. Capacitación.	Durante el desarrollo del proyecto.
6. Tamaño.	Nro. de personas involucradas en el proyecto.
7. Esfuerzo.	Horas por Personas estimadas y/o efectivas para la culminación del proyecto.
8. Rotación.	Por enfermedad, renuncia o despido.

**Tabla 3.16. Personal**

<b>Objetivos de negocio:</b> O1, O2, O3.	
Sub. Objetivo Nro. 3.	Conocer la utilidad de las herramientas para el desarrollo de SW.
Pregunta(s) Nro.	9, 10, 11.
<b>Entidad</b>	<b>HERRAMIENTAS.</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>

**Tabla 3.17. Herramientas**

<b>Objetivos de negocio:</b> O1, O2, O3.	
1. Tipo.	Sistema Operativo, Lenguaje de programación, Base de datos, etc.
2. Cantidad.	Nro. de computadoras, impresoras, etc.
3. Utilidad.	Conformidad del personal con las herramientas utilizadas.

**Tabla 3.17. Herramientas (continuación)**

<b>Objetivos de negocio:</b> O1, O2, O3.	
Sub. Objetivo Nro. 4.	Evaluar el impacto de documentar o no documentar los proyectos de software.
Pregunta(s) Nro.	49, 50, 51.
<b>Entidad.</b>	<b>DOCUMENTOS.</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Tipo	Todos los diferentes documentos producidos durante el proyecto.
2. Plantillas	Desarrolladas por la empresas, de estándares o de otras fuentes. (Si / No)
3. Esfuerzo	Horas por Persona empleadas en elaborar los diferentes tipos de documentos.
4. Volumen	Nro. de páginas por tipo de documento.

**Tabla 3.18. Documentos.**

<b>Objetivos de negocio:</b> O1, O2, O3.	
Sub. Objetivo Nro. 5.	Planificar con mejor precisión el proyecto.
Pregunta(s) Nro.	7, 13, 28, 33, 35, 8, 14, 23, 34, 36.
<b>Entidad</b>	<b>PROCESO DE DESARROLLO</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Tipo de aplicación de software.	Financiera, Contable, etc.
2. Tipo de proyecto de software.	Nuevo desarrollo, mejora, adaptación, etc.
3. Complejidad.	Estimada técnica y del negocio.

**Tabla 3.19 Proceso de desarrollo**

<b>Objetivos de negocio:</b>	O1, O2, O3.
4. Metodología o Modelo de proceso.	Desarrollada en la empresa o adquirida a terceros.
5. Inspecciones durante el proceso de desarrollo.	Revisiones y reportes de problemas.
6. Progreso.	Continuidad y cumplimiento de las tareas planificadas.
7. Tiempo.	Días, semanas o meses calendario.
8. Costo.	Estimado vs. Real.

**Tabla 3.19 Proceso de desarrollo (continuación)**

<b>Objetivos de negocio</b>	O1, O2, O3.
Sub. Objetivo Nro. 5.	Planificar con mejor precisión el proyecto.
Sub. Objetivo Nro. 6.	Monitorear los cambios en los requerimientos durante el desarrollo del proyecto.
Pregunta(s) Nro.	37, 38, 39.
<b>Entidad</b>	<b>REQUERIMIENTOS</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Cantidad de requerimientos a desarrollar.	Definidos en las fases iniciales del proyecto.
2. Cambios durante el desarrollo del proyecto.	Nuevos, borrados, modificados.

**Tabla 3.20. Requerimientos.**

<b>Objetivo de negocio</b>	O1, O3
Sub. Objetivo Nro. 5.	Planificar con mejor precisión el proyecto
Sub. Objetivo Nro. 7.	Generar diseños confiables.
Pregunta(s) Nro.	40, 42.
<b>Entidad.</b>	<b>DISEÑO</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Tipo.	Orientado a objetos, estructurado, etc.
2. Tamaño.	Nro. Clases, Nro. Casos de uso.
3. Nro. de inspecciones.	Se acepta, se rechaza o se piden modificaciones.

**Tabla 3.21 Diseño.**

<b>Objetivo de negocio</b>	O1, O2, O3
Sub. Objetivo Nro. 5.	Planificar con mejor precisión el proyecto
Sub. Objetivo Nro. 8.	Disminuir el tiempo de codificación del SW
Pregunta(s) Nro.	43, 44, 45
<b>Entidad</b>	<b>CÓDIGO FUENTE</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Estándar de codificación.	Definido y utilizado. (Si / No)
2. Complejidad.	De los métodos, funciones y procedimientos.
3. Re-utilización por componentes.	Re-utilizados, nuevos, borrados, modificados.

**Tabla. 3.22 Código fuente.**

<b>Objetivos de negocio</b>	O1, O2, O3
Sub. Objetivo Nro. 5.	Planificar con mejor precisión el proyecto.
Sub. Objetivo Nro. 6.	Monitorear los cambios en los requerimientos durante el desarrollo del proyecto.
Sub. Objetivo Nro. 7.	Generar diseños confiables.
Sub. Objetivo Nro. 8.	Disminuir el tiempo de codificación del SW.
Sub. Objetivo Nro. 9.	Disminuir la cantidad y gravedad de los defectos durante el desarrollo del SW y en el producto final.
Pregunta(s) Nro.	27, 46.
<b>Entidad</b>	<b>DEFECTOS / FALLAS</b>
<b>Atributos</b>	<i>Descripción de los atributos.</i>
1. Cantidad.	Defectos descubiertos * Inspección y/o en las pruebas.
2. Tipo.	"origen", (requerimientos, diseño, codificación, etc.)
3. Severidad.	Leve, grave, crítica.
4. Tiempo en cola.	Tiempo transcurrido desde que se generó el reporte de defectos.
5. Tiempo de corrección.	Tiempo que se requiere para corregir.
6. Estatus.	Abierto, cerrado, descartado.
7. Fallas.	Defectos que se presentan después de liberar el software.

**Tabla. 3.23. Defectos / Fallas**

### 3.3.5. Objetivos de medición

En esta paso, se construye de forma estructurada los objetivos de medición con el propósito de evitar pasar por alto puntos importantes en la definición y recopilación de datos.

<b>Sub. Objetivo Nro. 1</b>	
Conocer el nivel de satisfacción del cliente y su participación en el proyecto.	
<b>Objetivo de medición Nro. 1</b>	
<b>Objeto de interés:</b>	El proceso de desarrollo del software
<b>Propósito:</b>	Determinar
si	El nivel de satisfacción del cliente esta relacionado con su participación en el proyecto
para	Limitar o promover su colaboración con el personal asignado al proyecto.
<b>Perspectiva:</b>	
Examinar la:	Satisfacción del cliente
Desde el punto de vista del:	Administrador de proyectos
<b>Entorno y restricciones</b>	Las empresas por lo general tienen definidos sus propios mecanismos de evaluación de satisfacción del cliente.
	El desarrollo se realiza en unos casos dentro y/o fuera de la empresa de SW.
	Es aconsejable tomar en cuenta las principales características de un producto de software.

**Tabla 3.24. Objetivo de medición Nro. 1.**

### 3.3.6. Preguntas cuantificables e indicadores

Se procede a elaborar un conjunto de preguntas, a través de las cuales se llegue a saber si se logran los objetivos que hemos planteado. Además, se presenta uno de los indicadores asociados a estas preguntas.

<b>Objetivo de medición Nro. 1</b>	Determinar si el nivel de satisfacción del cliente está relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.
Pregunta Nro. 1	¿Cuál es el nivel de satisfacción del cliente y/o usuario con el producto de software desarrollado?
Pregunta Nro. 2	¿Cuál es la percepción del cliente con la metodología de desarrollo de proyectos de la empresa?
Pregunta Nro. 3	¿En que actividades durante el desarrollo del proyecto participó el cliente y/o usuario final?

Indicador: Clientes satisfechos vs. Participación en el proyecto.

Esfuerzo relativo	Satisfacción del Usuario	Satisfacción del Cliente
5,0%	3,7	3,1
9,1%	3,0	4,0
19,4%	4,5	4,8

Tabla 3.25. Preguntas cuantificables e indicador

### 3.3.7. Datos a recopilar

El objetivo de este paso, consiste simplemente en identificar para todos los indicadores que estamos interesados en generar los datos que necesitaremos para hacerlo. En nuestro caso de ejemplo, en la

tabla 3.26 se listan los datos que conforman el indicador “Clientes Satisfechos vs. Participación en el proyecto”.

<b>Indicador: “Clientes Satisfechos vs. Participación en el proyecto”</b>
Proyecto, (Nombre o código)
Satisfacción del cliente.
Satisfacción del usuario.
Fecha de fin del proyecto.
Esfuerzo real del cliente / usuario empleado en el proyecto.

**Tabla 3.26. Datos necesarios para formar el indicador**

### **3.3.8. Definición de los datos de las mediciones**

En este apartado, se definen los datos que se identificaron en el paso anterior, con el propósito de mejorar el entendimiento de que es lo que se desea registrar. Adicionalmente se elaboran los documentos que contienen las guías a seguir para el correcto ingreso de ciertos datos que pudieran resultar confusos a quien no este familiarizado con ellos. La tabla 3.27 muestra las referencias hacia, los documentos de definición de datos y guías de ingreso, aunque en este caso, los datos necesarios para el indicador “Clientes satisfechos vs. Participación en el proyecto”, solo requiere la definición de los mismos.



DATOS	Referencia a: Definición de datos (DF) y Guías para reportar (GR).
Proyecto	DF (1)
Satisfacción del cliente	DF (6)
Satisfacción del usuario	DF (7, 8)
Fecha fin	DF (10)
Esfuerzo real	DF (30)

**Tabla 3.27. Referencia a la definición de los datos a recopilar.**

**DF (1). Proyecto.** Un conjunto administrado de recursos interrelacionados que entrega uno o más productos a un consumidor o usuario. Este conjunto de recursos tiene inicio y fin definidos y generalmente operan de acuerdo a un plan. Tal plan es frecuentemente documentado y especifica el producto a ser entregado o implementado, los recursos y fondos usados, el trabajo a ser realizado y el calendario para realizar dicho trabajo. **[28]**

**DF (6). Cliente.** La persona u organización que paga por el producto y usualmente (pero no necesariamente) decide los requerimientos. **[29]**  
La persona u organización que desea usar el servicio, producto o resultado del proyecto. **[30]**

**DF (7). Usuario final.** Persona o personas, que operan o interactúan directamente con el sistema. **[31]**

**DF (8). Satisfacción del usuario.** La percepción del usuario de haber utilizado correctamente su tiempo, habiendo recibido respecto de sus propias expectativas y en un determinado contexto ambiental la mejor prestación posible del servicio o producto adquirido. **[32]**

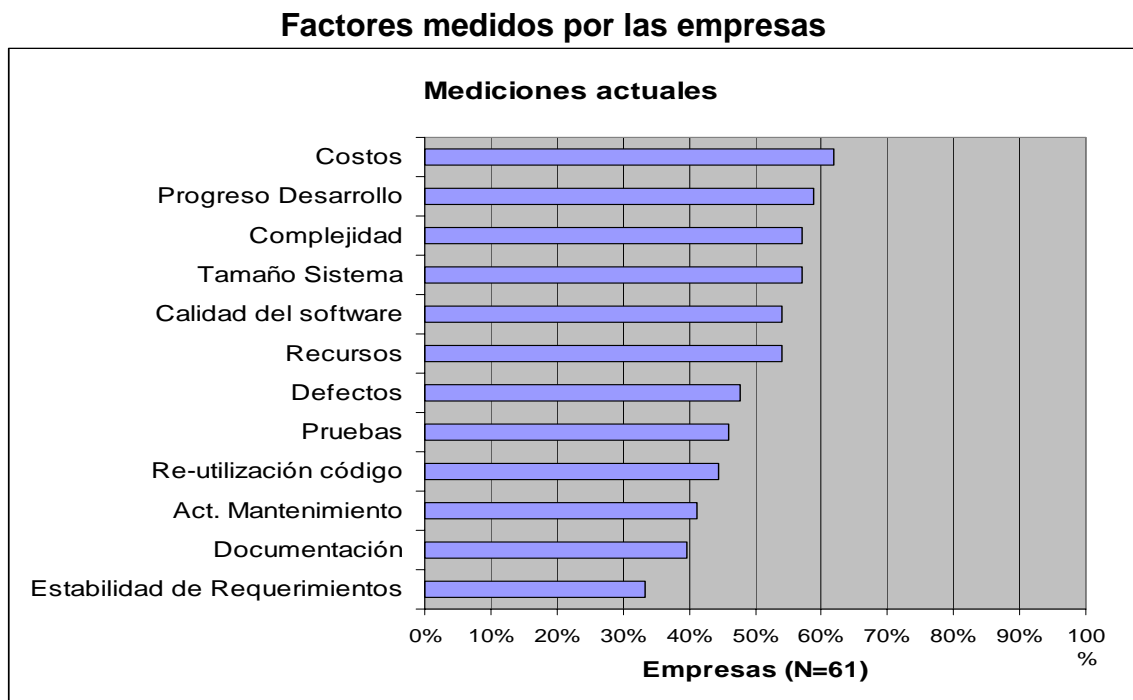
**DF (10). Fecha de terminación del proyecto.** La fecha (día, mes y año) cuando la empresa de software ha realizado la entrega formal del producto de software que ha desarrollado. **[33]**

**DF (30). Esfuerzo.** El número de horas requeridas para completar una tarea, un entregable, una fase o un proyecto. El enfoque aquí es sobre la cantidad de tiempo expresado en horas. **[24]**

### **3.3.9. Análisis, diagnóstico y acción**

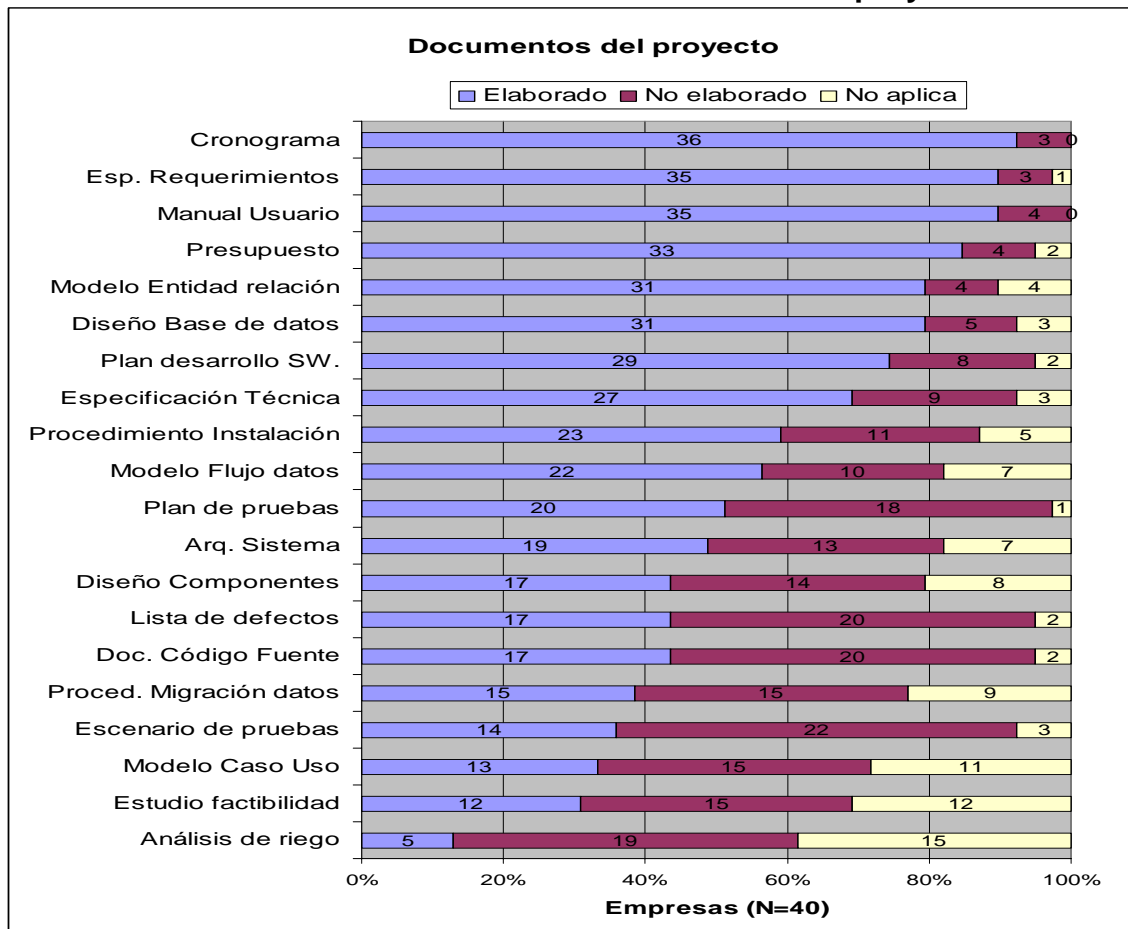
En el contexto de nuestra tesis, la actividad de **análisis** comprende la búsqueda de mediciones que se realicen actualmente en las empresas de software ecuatorianas. El estudio exploratorio ayudó con esta tarea cuyos resultados se pueden apreciar en el gráfico 3.2, además el gráfico 3.3 complementa estos datos para concluir que:

- 1) La medición de factores que afectan la gestión de un proyecto de software es una actividad que no se realiza en todas las empresas evaluadas, y
- 2) Durante el proceso de desarrollo de software no todas las empresas elaboran todos los documentos que pudiesen ser fuente de los datos necesarios para las métricas.



**Figura 3.2.**

## Documentos elaborados durante el desarrollo de proyectos



**Figura 3.3.**

Para el **diagnóstico** evaluamos los resultados de la actividad de análisis con el propósito de determinar de qué manera se podrá implementar esta iniciativa de medición de los proyectos de software. El principal problema que impide realizar una adecuada documentación (según el estudio exploratorio), es la premura por terminar el proyecto de software, por lo cual se debe considerar los siguientes aspectos:

- 1) Los datos a medir deben ser los considerados importantes para la mayoría de las empresas, y
- 2) Los medios de recopilación de estos datos deben ser sencillos, concisos, y en lo posible, compatibles con las prácticas actuales de la empresa.

El haber seguido el proceso de medición de software dirigido a objetivos nos ha ayudado a cubrir el primer punto, pues ya contamos con un conjunto de datos seleccionados que fueron descritos y definidos en los pasos 7 (Datos a recopilar) y 8 (Definición de datos) respectivamente.

Por otro lado, el 76% de las empresas evaluadas en el estudio exploratorio indicaron que el Microsoft Project es una de las herramientas que utilizan para la planificación. Esto indica que el proyecto es desglosado en tareas a las que se les asigna los recursos necesarios para el desarrollo de las mismas dentro de un tiempo determinado. Entonces, resulta conveniente orientar el diseño de los instrumentos de recopilación de datos a aprovechar esta situación y de esta forma cubrir el punto 2.

Para terminar el diagnóstico, los datos que nos interesan recopilar deben ser evaluados para determinar la disponibilidad de los mismos, es decir, determinar la facilidad con la cual se los podría obtener. También es necesario identificar las herramientas (encuestas, formularios, bases de datos, etc.) o documentos (presupuestos, cronogramas, lista de requerimientos, etc.) en donde se los pueda encontrar. Para ilustrar este punto, continuamos con nuestro indicador de ejemplo “**Cientes Satisfechos vs. Participación en el proyecto**” y en la tabla 3.28 mostramos los datos necesarios para su formación junto con la disponibilidad y fuente de consulta de esos datos.

<b>Formulario (Código)</b>	<b>Datos</b>	<b>Disponibilidad</b>	<b>Fuente(s)</b>
<b>F00(1)</b>	Proyecto	Obtenida con poco esfuerzo	Formulario F00, y/o la orden de trabajo.
<b>F00(5)</b>	Satisfacción del cliente	Obtenida con poco esfuerzo	Formulario F00, y/o encuesta aplicada al cliente.
<b>F00(6)</b>	Satisfacción del usuario	Obtenida con poco esfuerzo	Formulario F00, y/o encuesta aplicada al usuario.
<b>F00(8)</b>	Fecha fin	Disponible	Formulario F00, y/o MS. Project.
<b>F01(39)</b>	Esfuerzo real	Disponible	Formulario F01, y/o MS. Project.

**Tabla 3.28. Disponibilidad de Datos**

Con los resultados del análisis y diagnóstico que acabamos de realizar, finalmente pasamos a la **acción**, que es la actividad en la cual

se determinan las actividades a realizar y se construyen los medios a utilizar para llevar a la práctica todo lo desarrollado hasta ahora.

Como ejemplo de esto, la figura 3.4 muestra el diseño del formulario F00 que se utilizará para el registro y consulta de los datos requeridos para formar el indicador “**Cientes Satisfechos vs. Participación en el proyecto**”.

### Formulario F00. Datos generales del proyecto

DATOS GENERALES DEL PROYECTO (F00)			
PROYECTO		REQUERIMIENTOS	
1		2	
TIPO DE:		SATISFACCION	
Aplicación de software	3	Cliente	Usuario
Proyecto de software	4	5	6
FECHA		PERSONAL	
Inicio	Fin	Estimado	Real
7	8	9	10
COSTO		COMPLEJIDAD	
Estimado	Real	Negocio	Técnica
11	12	13	14
ROL - NOMBRE		EXPERIENCIA	
15	16	Desarrollo	Negocio
		17	18
ROTACION		INTERRUPCIONES	
19		Tiempo	Causa
		20	21

Figura 3.4

### **3.3.10. Plan de medición**

Con los objetivos de medición ya establecidos (paso 5), los indicadores (gráficos o tablas) que se desean trazar bosquejados (paso 6), los datos que se medirán ya definidos (7, y 8) y las guías e instrumentos que nos ayudaran a registrarlos ya construidos (paso 9). El paso final en el proceso de “medición de software basado en objetivos” es la formación del plan de medición (ver apéndice B) integrando todas estas partes desarrolladas por separado en cada uno de los pasos anteriormente descritos.

## **3.4. APLICACIÓN EN TRES EMPRESAS DEL MEDIO**

### **3.4.1. Forma de aplicación**

Para la aplicación del plan de medición, solicitamos la colaboración de 3 empresas de software, las cuales escogieron un proyecto de desarrollo de software disponible para la aplicación de esta herramienta.

En cada una se procedería a coordinar reuniones para explicar la forma como se formó el plan y la manera cómo utilizar los instrumentos de recopilación de datos dentro del proceso de desarrollo



de software, actividad que se estimó se realizaría en 3 sesiones y en las que se les proporcionaría los materiales desarrollados.

### **3.4.2. Facilidades y dificultades en la aplicación**

Entre las facilidades que tuvimos en las empresas de software que aplicaron el plan de medición podemos mencionar las siguientes:

- 1) Existió interés por cooperar en la realización del estudio exploratorio y por consiguiente concedieron el tiempo requerido para la descripción y explicación del plan de medición. Actividad realizada en 3 reuniones de 1 hora cada una.
  
- 2) Ciertas prácticas existentes en las empresas como por ejemplo el reporte de actividades del personal eran compatibles y facilitaban el uso de nuestros formularios.

No obstante, a pesar de nuestros esfuerzos por generar una herramienta que sea fácil de usar, que esté acorde a las prácticas de la mayoría de las empresas de software ecuatorianas y a pesar del interés y compromiso inicial de las empresas que aplicaron el plan de medición, se presentaron las siguientes dificultades:

- 1) El personal de las empresas fue instruido de registrar los datos que se necesitaban pero en la medida que esto no los retrasara de sus tareas habituales.
- 2) La descripción del plan de medición solo la recibió uno de los miembros del equipo asignado al proyecto, él que a su vez replicaría esta información a los demás miembros.
- 3) Hubieron ciertas restricciones con algunos datos del proyecto, especialmente los referentes al costo del mismo.

CAPITULO

4

Resultados

# **CAPITULO 4**

## **RESULTADOS**

### **4.1. RESUMEN DE LAS CARACTERÍSTICAS PRINCIPALES DE LAS EMPRESAS DE SOFTWARE**

En este apartado, se describen aspectos generales de las empresas evaluadas referentes a su distribución geográfica, tamaño, años de funcionamiento, y exportación de sus productos de software. También se presentan los resultados obtenidos de la aplicación del estudio exploratorio referente a la forma de trabajar de las empresas de software ecuatorianas.

#### **4.1.1. Aspectos generales sobre las empresas de software**

##### **4.1.1.1. Ubicación y tamaño de las empresas**

Sesenta y tres (63) empresas fueron encuestadas, el 79% de éstas están localizadas en Quito y 21% restante en Guayaquil. Para clasificar a estas empresas según su tamaño, emplearemos el criterio de clasificación “número de empleados”, obteniendo los resultados de la tabla 4.1

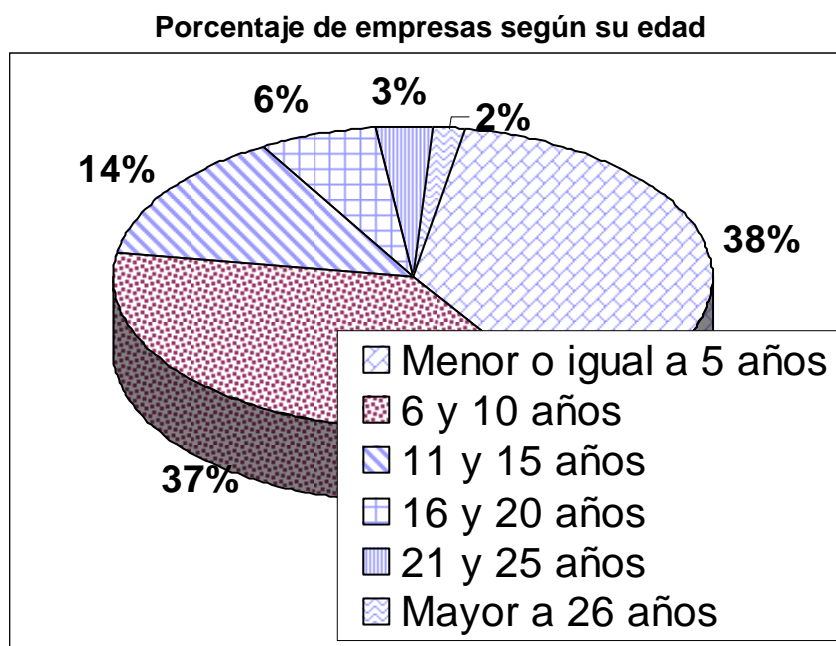
<b>Número de empleados</b>	<b>Empresas (N = 63)</b>		<b>Tamaño</b>
Menor o igual a 10	26	41%	Pequeña
Entre 10 y 50	35	56%	Mediana
Más de 50	2	3%	Grande

**Tabla 4.1 Tamaño de las empresas**

Este criterio de clasificación es similar al usado en el estudio “El sector de software y servicios informáticos” [27] que consideró a empresas con 10 o menos empleados como “pequeñas”, empresas entre 10 y 50 empleados como “medianas” y empresas con más de 50 empleados como “grandes”.

#### **4.1.1.2. Años de funcionamiento**

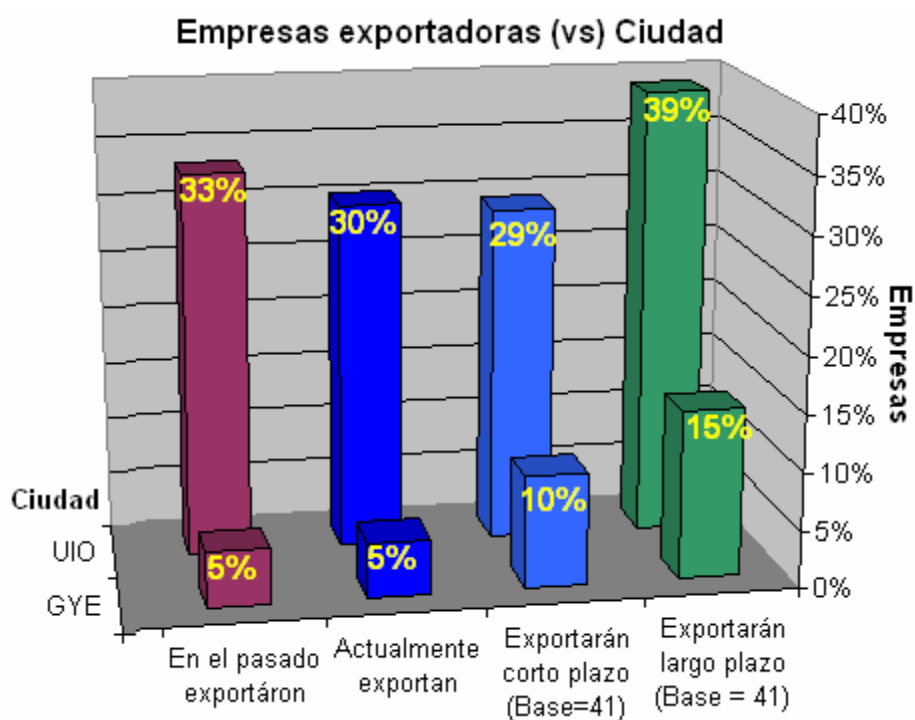
Hasta el año 2005, el tiempo promedio de funcionamiento de las empresas evaluadas fue de 8 años y sólo el 25% de ellas superaron los 10 años en el mercado. La figura 4.1 muestra en porcentajes la edad de las empresas.



**Figura 4.1**

#### **4.1.1.3. Empresas que exportan software**

Hasta el 2005, el 35% de las empresas exportaban software, de éstas, el 30% estaba localizado en Quito, el restante 5% en Guayaquil. Para las empresas que exportan software, este rubro representa en promedio el 37% del valor facturado y el 48% del software desarrollado. De las 41 empresas que no exportan, el 39% espera hacerlo en el corto plazo (UIO 29%, GYE 10%). y el 54% en el largo plazo (UIO 39%, GYE 15%).



**Figura 4.2**

Solamente 6 empresas (10% de 63) tienen la certificación ISO 9001:2000 y todas se encuentran o tienen su sede principal en la capital del Ecuador.

#### **4.1.2. Aspectos generales de la gestión de proyectos de software**

En este apartado se presentan los valores observados en diversos factores que influyen en las diferentes actividades que comprenden un proyecto de software.

#### 4.1.2.1. Metas de desarrollo del proyecto

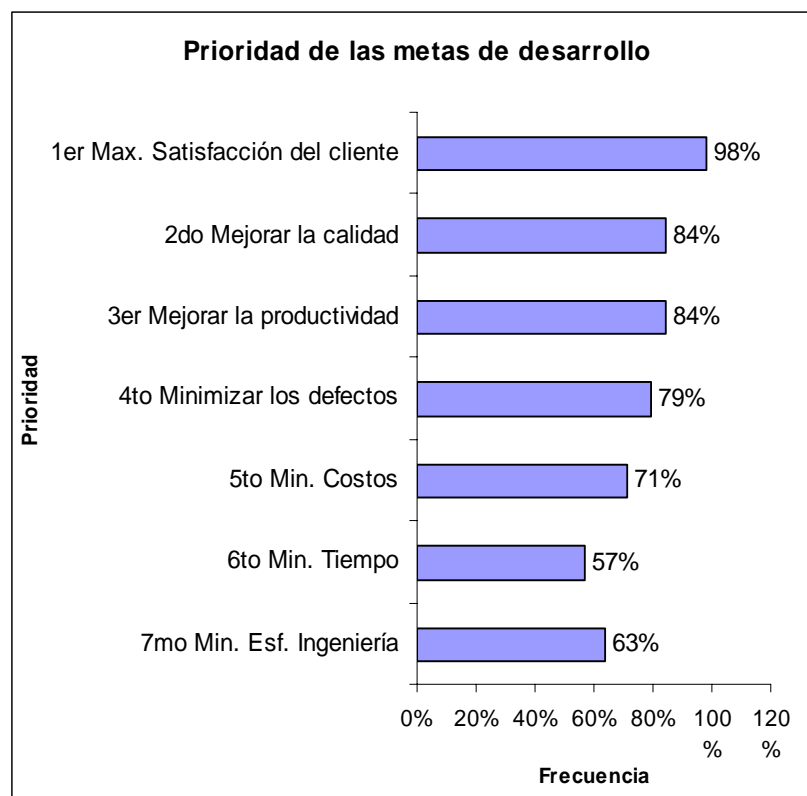
De forma general se puede mencionar que los objetivos de negocio más importantes que tienen las 63 empresas evaluadas al momento de desarrollar proyectos de software son:

Objetivos de negocio	Prioridad
Maximizar la <b>satisfacción del cliente</b>	1ero según el 98% de empresas
Mejorar la <b>calidad</b> del producto de software	2do según el 84% de empresas
Mejorar la <b>productividad</b>	3ero según el 84% de empresas

**Tabla 4.2 Metas de desarrollo.**

Otros objetivos que formaron parte de esta evaluación se observan en la figura 4.3:



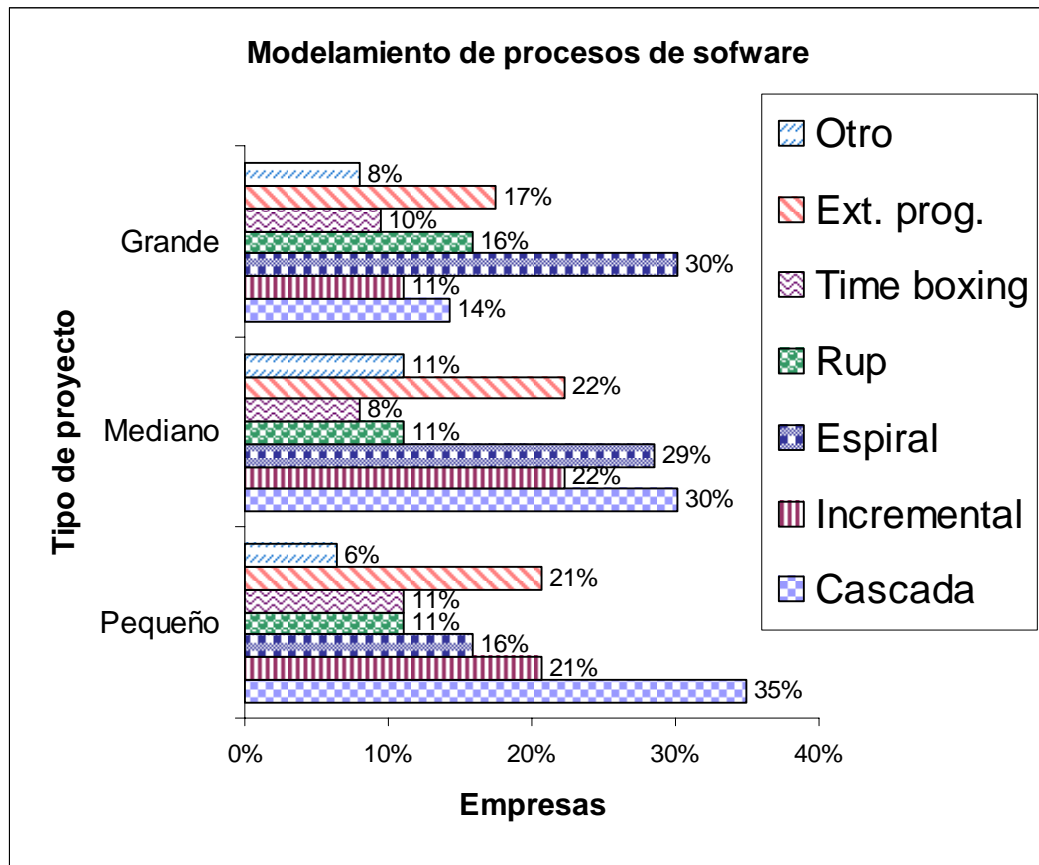


**Figura 4.3**

#### 4.1.2.2. Modelos y metodologías usadas

La forma de evaluación de los modelos y metodologías del proceso de desarrollo que utilizan las empresas está relacionada con la duración de los proyectos de software que para efectos de nuestro estudio se la definió de la siguiente forma: 1. Proyectos pequeños (consumen entre 1 y 6 meses), 2. Proyectos medianos (entre 7 y 15 meses) y 3. Proyectos grandes (más de 15 meses). Teniendo esto en cuenta, en la figura 4.4 se observa que el modelo cascada es más utilizado

en proyectos pequeños, mientras que el modelo espiral se utiliza en proyectos grandes.



**Figura 4.4**

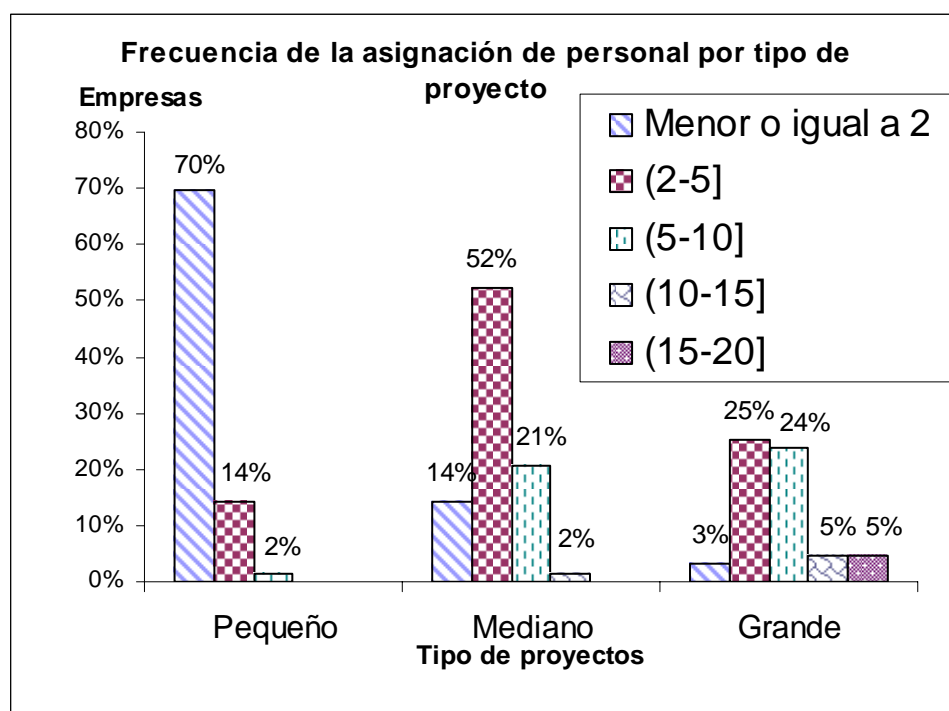
#### 4.1.2.3. Documentación del proyecto

El 40% de las empresas siempre documentan el proceso de desarrollo del software y entre las causas que desmotivan a las empresas para no documentar se encontró la premura para

terminar los proyectos (78% de los casos) y en menor medida el poco interés en realizar esta actividad (14% de los casos).

#### 4.1.2.4. Tamaño del personal asignado a los proyectos de software

El 70% de las empresas asignan hasta 2 personas para los proyectos pequeños. El 52% asignan entre 3 y 5 personas para proyectos medianos, el 25% asignan entre 3 y 5 personas para proyectos grandes y el 24% asigna entre 6 y 10 personas. En la figura 4.5 se observan más resultados:



**Figura 4.5**

#### 4.1.2.5. Distribución del tiempo y costo en los proyectos de software

Para obtener estos valores, el proceso de desarrollo se lo dividió en 4 fases siguiendo el ciclo de vida del software pero excluyendo la fase de operación y mantenimiento pues la tesis se enfoca en el desarrollo. En promedio las fases que consumen más tiempo en su orden son: En primer lugar la codificación, en segundo lugar el diseño, en tercer lugar la especificación y requerimientos, por último la integración y pruebas del sistema.

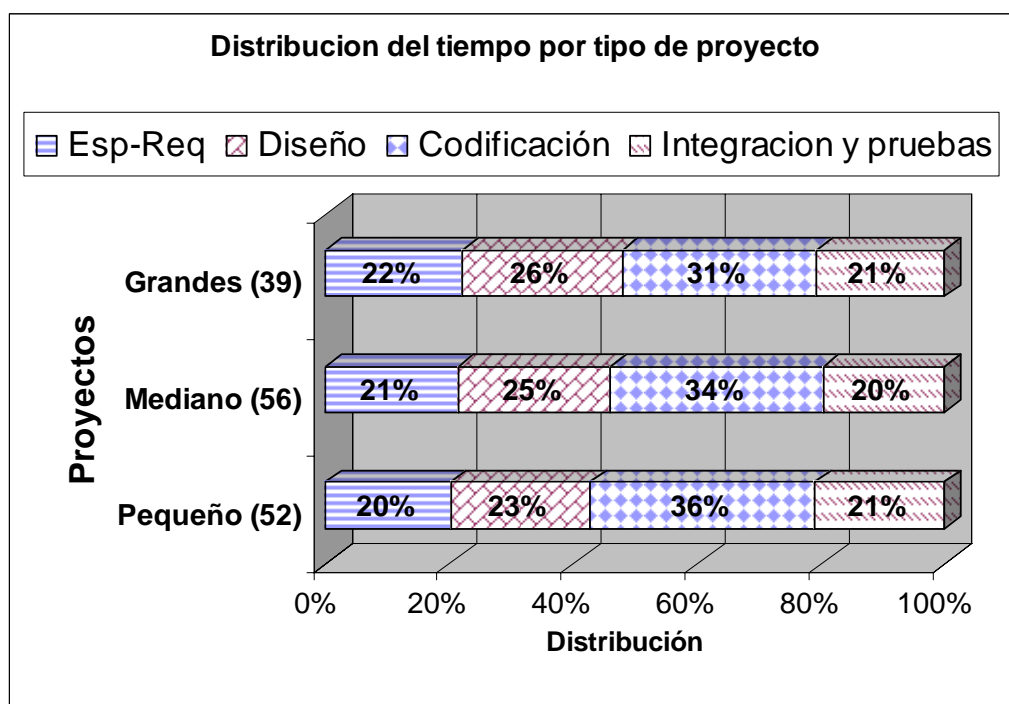
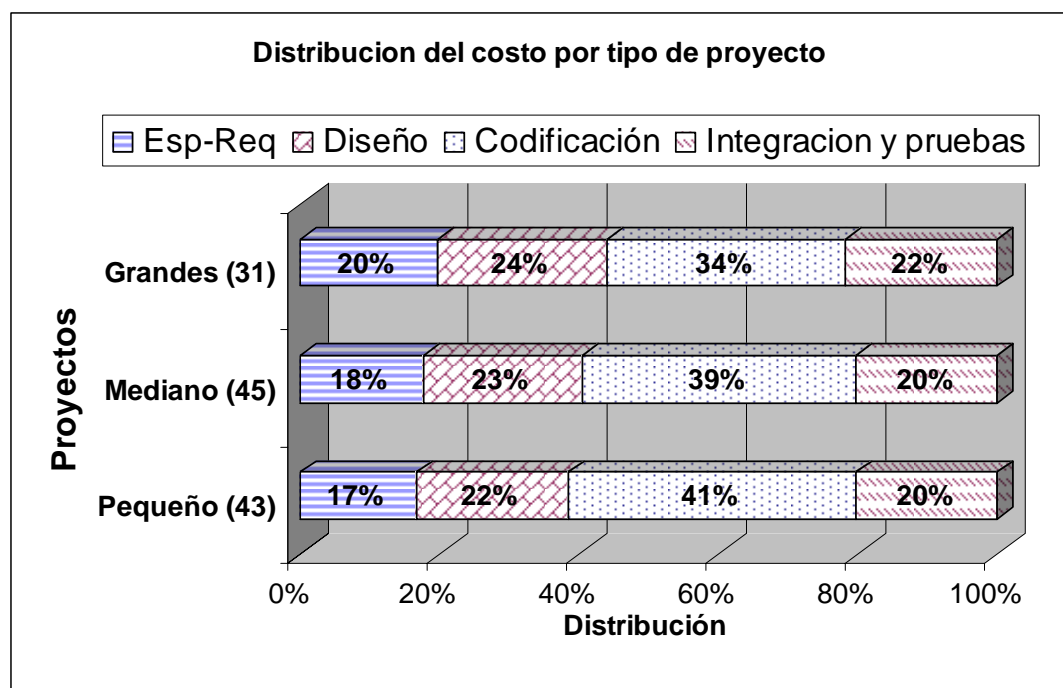


Figura 4.6

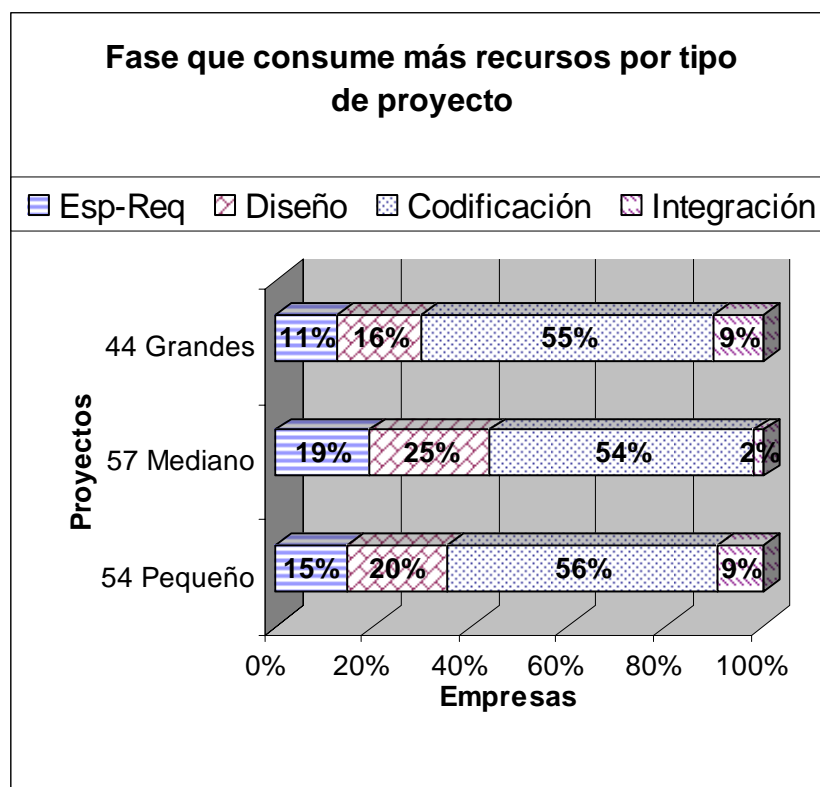
La fase que consume la mayor cantidad de tiempo es la codificación y prueba de unidades, sin importar el tamaño del proyecto. Una situación similar tiene la distribución del costo, la cual se observa en la figura 4.7



**Figura 4.7**

Consistente con lo mencionado anteriormente en la figura 4.8 la distribución de los recursos del proyecto se concentra en la fase de codificación y pruebas sin importar el tamaño de proyecto.

El 63% de las empresas llevan un registro del tiempo que toman las diferentes actividades del desarrollo siendo las más señaladas el diseño (60%), y la codificación (54%).

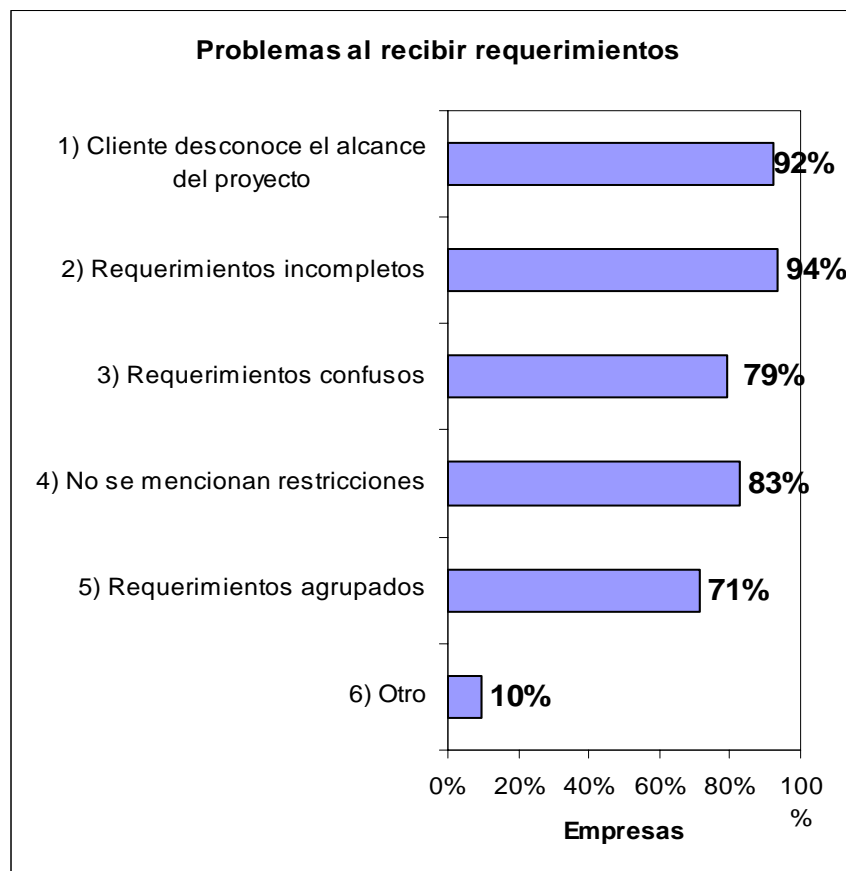


**Figura 4.8**

#### **4.1.2.6. Problemas frecuentes con los requerimientos del usuario**

Los dos problemas más comunes al momento de recibir las especificaciones y requerimientos del usuario son: 1) requerimientos incompletos (94% de las empresas), y 2)

desconocimiento del alcance del proyecto por parte del cliente (92% de las respuestas). En la figura 4.9 se observa la distribución de los problemas más comunes que se presentan en la fase de requerimientos.

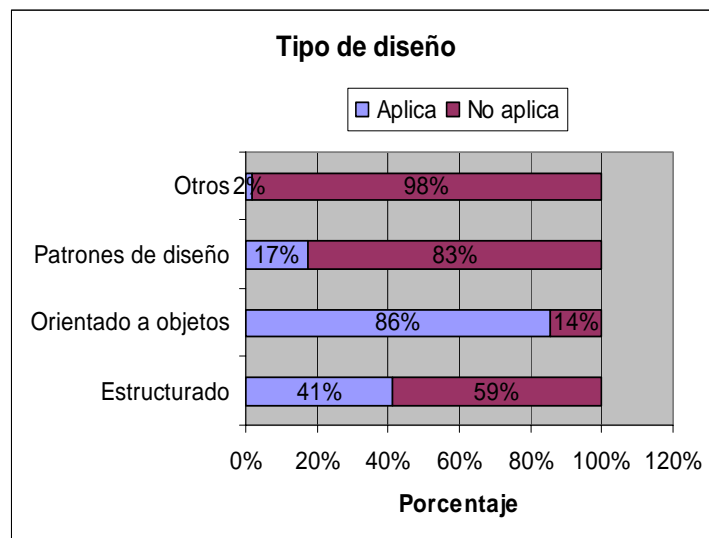


**Figura 4.9**

#### **4.1.2.7. Tipos de diseño de software utilizados**

El diseño orientado a objetos es ampliamente utilizado por las empresas, el diseño estructurado también es aceptado en un

porcentaje significativo, pero muy pocas empresas utilizan patrones de diseño como se puede observar en la figura 4.10.

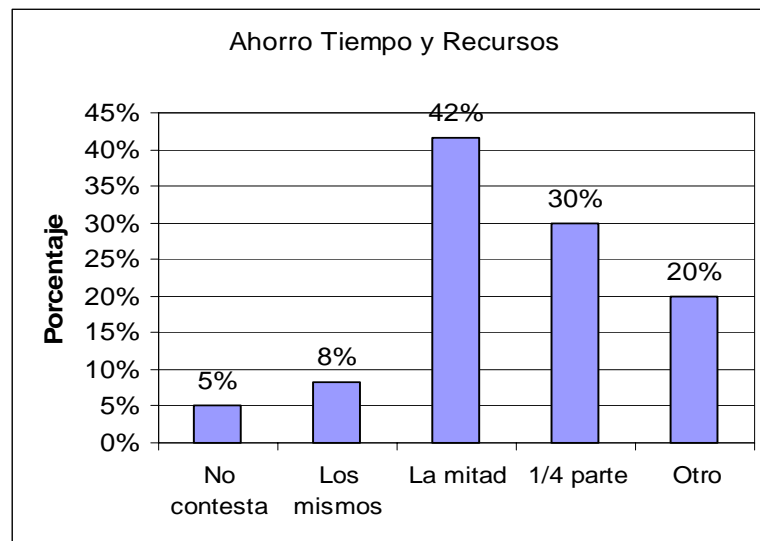


**Figura 4.10**

#### 4.1.2.8. Re-Utilización de componentes

El 95% de las empresas reutilizan los componentes de software desarrollados, de éstas, el 42% estiman que la reutilización les permite ahorrar la mitad de tiempo y recursos.

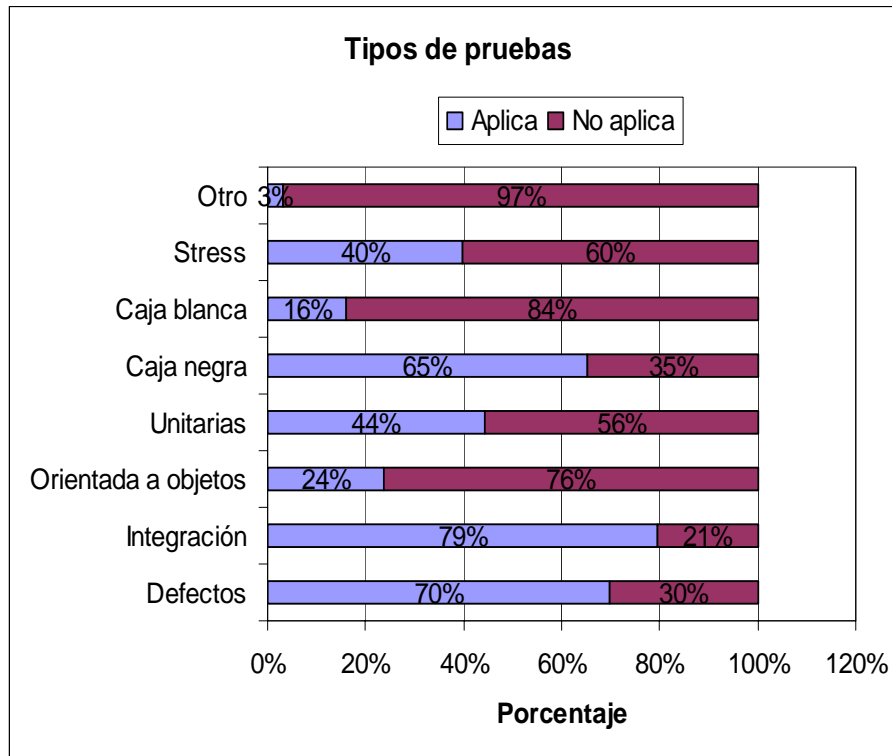




**Figura 4.11**

#### **4.1.2.9. Tipos de pruebas realizadas**

Respecto al tipo de pruebas que se realizan, las más utilizadas son las de integración (79%), de defectos (70%) y de Caja negra (65%). El 67% de los entrevistados afirman que las pruebas son realizadas principalmente con el cliente, un 56% realiza pruebas con su equipo de desarrollo y un 48% utiliza personal dedicado exclusivamente a pruebas. La figura 4.12 resume los resultados descritos. El 60% de las empresas prefieren hacer un control de defectos sólo cuando éstos se presentan. Al crecer el volumen de defectos encontrados las empresas comúnmente realizan dos acciones de contingencia: Modificar el tiempo planificado (68%) y/o asignar más personal al desarrollo del proyecto (40%).



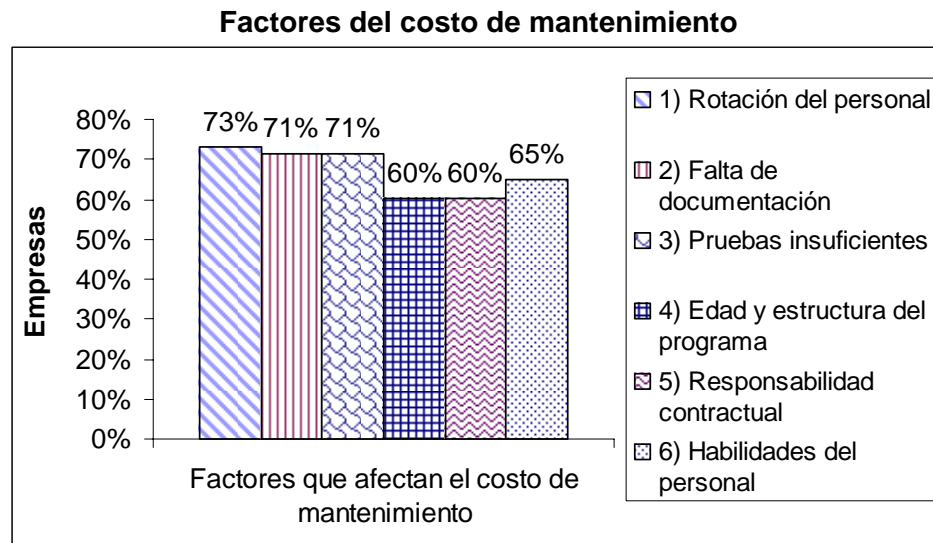
**Figura 4.12**

#### **4.1.3. Factores que aumentan el costo del mantenimiento del software**

Los tipos de mantenimiento de software más comunes sin importar el tamaño del proyecto son: el “perfectivo” (aquel que consiste en agregar funcionalidad al software) y el “correctivo” (aquel que consiste en corregir fallas en el software).

Los tres factores principales que aumentan el costo de mantenimiento son: 1) la rotación del personal (73%), 2) la falta de documentación

(71%), y 3) las pruebas insuficientes (71%). En la figura 4.13 se puede observar la puntuación obtenida por estos y otros factores que se tomaron en cuenta:



**Figura 4.13**

#### **4.2. PLAN PILOTO DE MEDICIÓN APLICADO A 3 EMPRESAS DE SW**

A continuación se mencionarán las características principales y los requisitos necesarios para la implementación del plan de medición elaborado durante esta tesis. Adicionalmente se mencionan sus limitaciones y algunas recomendaciones.

#### **4.2.1. Características principales**

Las métricas que forman parte de este plan, están formadas por 1) *objetivos de medición* que indican el propósito de la actividad de medición, 2) *preguntas* que reflejan la necesidad de información, 3) *indicadores* que resumen los resultados obtenidos y 4) los *datos* a ser medidos y recopilados.

Los instrumentos de medición están basados en plantillas que son fáciles de comprender e implementar en un proyecto de software basándose en las tareas que realiza el equipo de desarrollo.

#### **4.2.2. Requisitos necesarios para la implementación del plan de métricas**

Para la recopilación, resumen y presentación de resultados de las métricas se requiere de la autorización y apoyo permanente de la gerencia de la empresa.

Adicionalmente, es imprescindible capacitar al personal de la empresa en el uso de los instrumentos de recopilación de datos.

Es necesario utilizar mecanismos que permitan manipular los datos obtenidos de forma automática, de esta manera se facilita la revisión y

diseño de diferentes vistas de la información contenida en estos datos. Para el efecto existen las tablas dinámicas de Excel o el software SPSS.

#### **4.2.3. Limitaciones**

El uso de este plan está limitado a las entidades y atributos seleccionados del proceso de desarrollo de software, objeto de este estudio. Los resultados obtenidos nos permiten conocer el estado actual de las prácticas que se utilizan en las empresas evaluadas. Con estos resultados, se podrá decidir objetivamente si es necesario implementar una mejora del proceso y el grado de mejoría que se desea alcanzar.

La recopilación de datos no es automática, sino que depende de la diligencia con la que el personal de la empresa realice esta actividad, por lo que debe estar altamente convencido y motivado de estar realizando una actividad que le ayudará a solventar y evitar repetir problemas que se presenten con frecuencia.

#### **4.2.4. Recomendaciones para la implementación**

Proporcionar oportuna retroalimentación a quienes son la fuente de los datos para validar periódicamente la información que se va obteniendo.

Considerar como actividad obligatoria el registro de los datos requeridos para la formación de métricas, inclusive cuando el proyecto se encuentre atrasado en su cronograma.

### **4.3. METRICAS NO APLICADAS**

Se incluyó en los diseños de los instrumentos de recopilación de datos el registro de ciertas métricas (Líneas de código y complejidad ciclomática) cuyo uso es sugerido en muchos de los libros que tratan el tema de métricas. No obstante su aplicación no fue posible por las razones que se explican en los siguientes apartados.

#### **4.3.1. Complejidad ciclomática**

Es una métrica desconocida en nuestro medio y para la cual se desconocía una forma automática de cálculo y recopilación en los diferentes componentes que forman las aplicaciones de software, por lo que su cálculo hubiese tenido que ser manual. Las empresas

consideraron que obtener esta métrica podría requerir esfuerzo extra de su personal y no estaban dispuestos a obtenerla.

#### **4.3.2. Conteo de líneas de código**

Una de las dificultades para la aplicación de esta métrica es la variabilidad de estilos de programación de cada uno de los programadores, lo cual se agrava por la frecuente rotación de personal.

Adicionalmente, las tres empresas evaluadas utilizan diferentes lenguajes de programación, por lo cual una comparación no era posible.

### **4.4. ANÁLISIS DE LAS MÉTRICAS APLICADAS DURANTE EL PLAN PILOTO**

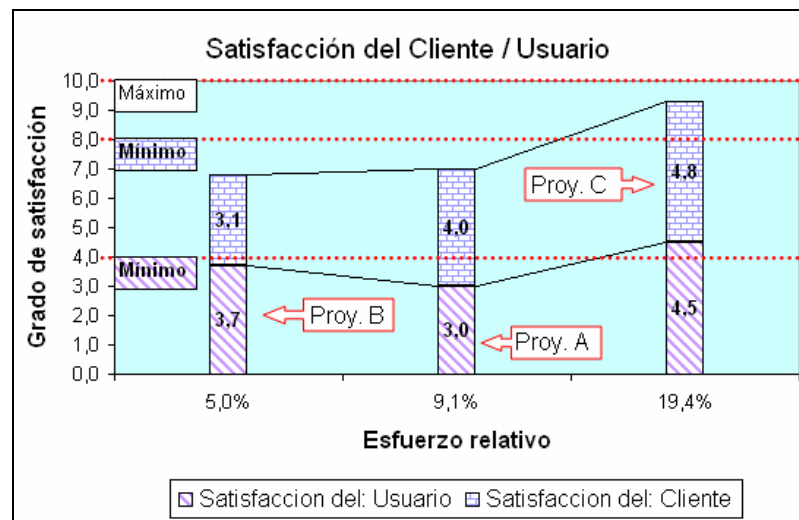
Por medio de gráficos y tablas (indicadores) resumiremos los datos obtenidos de los 3 proyectos evaluados.

#### **4.4.1. Satisfacción de los clientes / usuario**

Los valores de satisfacción obtenidos de los clientes/usuarios y su participación relativa durante el desarrollo el proyecto de forma general

indican que a mayor participación con el personal de la empresa mayor será el nivel de satisfacción que lograrán con el producto de software que la empresa les proporcione.

En la figura 4.14 se puede observar los diferentes valores de satisfacción y participación relativa de los clientes y usuarios durante el desarrollo del proyecto de software. Los valores de la escala vertical 4 y 8 son puntos de referencia que nos ayudan a clasificar a los proyectos que están dentro de un nivel aceptable de satisfacción o que no han alcanzado estos valores mínimos, pudiendo ser considerados como proyectos que no cumplen las expectativas de sus clientes/usuarios.





Como ejemplo de lo anteriormente mencionado los usuarios de los proyectos A y B no alcanzaron el valor mínimo 4. Por consiguiente, estos valores nos indican que estos proyectos tienen oportunidades de mejora que podrían ser aprovechadas y tomar acciones pertinentes, de igual forma con respecto a la satisfacción del cliente el proyecto B presenta un valor por debajo de lo aceptable mientras que el proyecto A consigue el valor 4 considerado (en nuestro criterio) el valor mínimo de satisfacción aceptable. Se menciona sin embargo que el valor que se considere mínimo para determinar que un proyecto no alcanza el nivel de satisfacción deseado depende de cada empresa. El proyecto C obtuvo valores de satisfacción aceptables sobrepasando el valor mínimo de 4.

El valor 10 en la escala vertical representa el valor máximo de satisfacción de los clientes y usuarios respecto de la metodología de desarrollo empleada y al producto de software obtenido.

#### **4.4.2. Esfuerzo del personal**

En los tres casos, de forma general, la mayor cantidad de esfuerzo fue consumido por el personal que desempeña el rol de desarrollador.

En el Proyecto A se contabilizó 3151 horas/hombre, de las cuales la mayor parte han sido consumidas por los siguientes roles: Developer Oracle con 1429 horas (45%), Developer .Net con 877 horas (27%) y Jefe Proyecto con 518 (16%). En el proyecto B se contaron 719 horas/hombre, el consumo de estas horas se distribuyó de la siguiente forma: Administrador con 15 horas (2%) y Desarrollador con 704 horas (98%). En el Proyecto C se contabilizó 141 horas/hombre y fueron consumidas totalmente por el desarrollador.

#### **4.4.3. Error de estimación**

Después de haber hecho la estimación del esfuerzo que una tarea en particular requiere se pueden presentar las siguientes situaciones: 1) Es posible cumplir esta tarea con la cantidad de esfuerzo determinada, 2) Se puede consumir menos esfuerzo del estimado o 3) Se empleará más esfuerzo del estimado. Pero para nuestro estudio nos interesa analizar el esfuerzo que se excede de lo estimado, ya que como se verá más adelante es el que se presenta con mayor frecuencia.

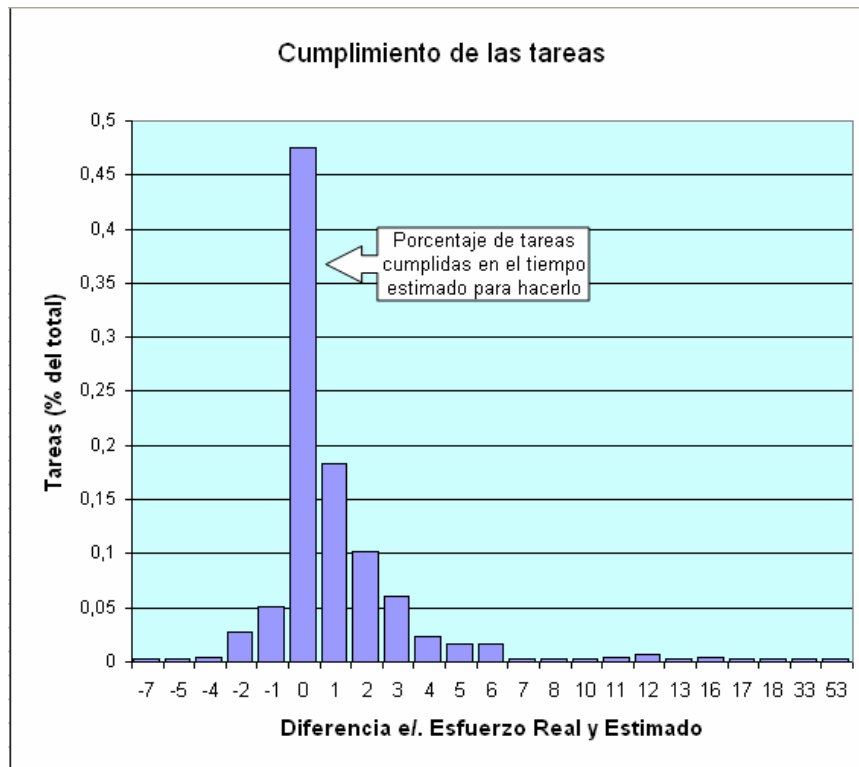
En el proyecto A, se registraron 431 tareas desde el 27 de Julio de 2006 hasta el 13 de Octubre del 2006 (período en el cual se aplicó el estudio), de estas tareas un 9% consumieron menos horas/hombre de las estimadas, mientras que el 48% consumieron exactamente las

horas/hombre estimadas, y el restante 43% consumieron más horas/hombre de las que fueron estimadas al inicio de cada una.

En la figura 4.15 se observan los datos mencionados anteriormente, siendo la escala horizontal la diferencia entre el esfuerzo real y el estimado de las tareas del proyecto y el eje vertical el porcentaje de las tareas que tienen las diferencias mostradas. Fácilmente se aprecia que la mayoría de tareas que excedieron el esfuerzo estimado (40%) están en el rango de 1 a 6 horas/hombre de diferencia.

El valor mínimo y máximo del esfuerzo estimado de las tareas es 1 y 51 horas por hombre respectivamente, pero en el intervalo de esfuerzo estimado entre 1 y 30 se aglutina el 99% de todas las tareas registradas.

En la figura 4.15, en el valor cero del eje horizontal se ubican las tareas que se cumplieron en el tiempo estimado. Hacia la derecha se muestran las tareas que emplearon más esfuerzo del estimado.

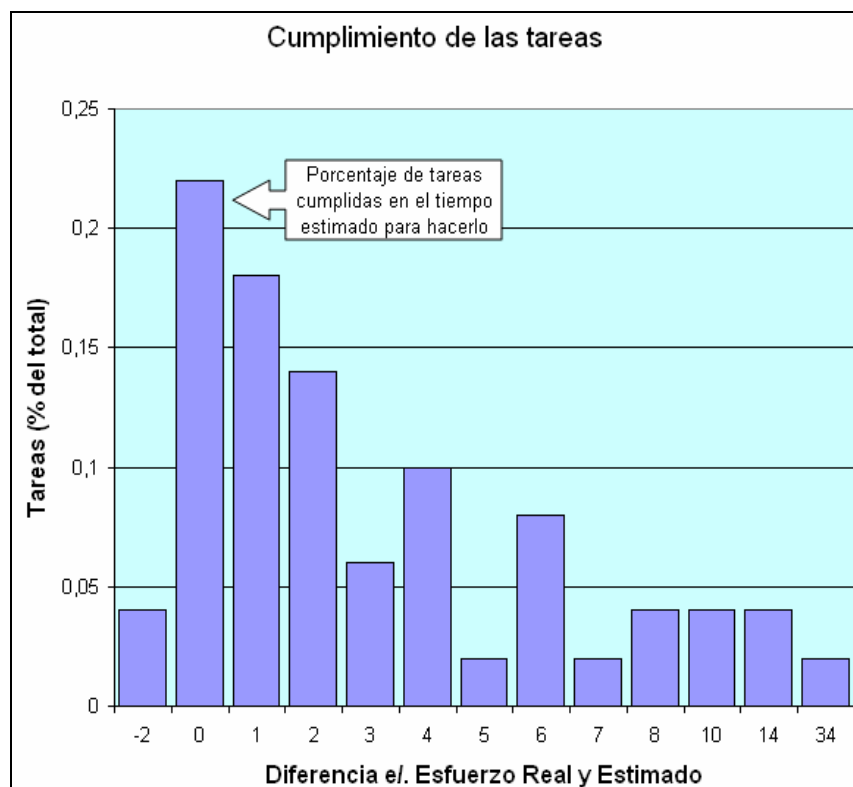


**Figura 4.15**

Para separar valores muy grandes que pudiesen distorsionar el cálculo del error de estimación promedio, se escogen las tareas con un esfuerzo estimado entre 1 y 30 horas x hombre (99% de las tareas) y con una diferencia entre 1 y 6 horas /hombre entre el esfuerzo real y el estimado (40% de las tareas con un esfuerzo real mayor al estimado). De esta forma se determina que el error promedio en la estimación de las tareas del proyecto A es: 39.7%

En el proyecto B, se registraron 50 tareas desde el 10 de Julio de 2006 hasta el 5 de Octubre del 2006 (período en el cual se aplicó el

estudio), de estas tareas un 4% consumieron menos horas/hombre de las estimadas, mientras que el 22% consumieron exactamente las horas/hombre estimadas, y el restante 74% consumieron más horas/hombre de las que fueron estimadas al inicio de cada una. En la figura 4.16 se observan los datos mencionados anteriormente. Fácilmente se aprecia que la mayoría de tareas que excedieron el esfuerzo estimado (58%) están en el rango de 1 a 6 horas/hombre de diferencia.

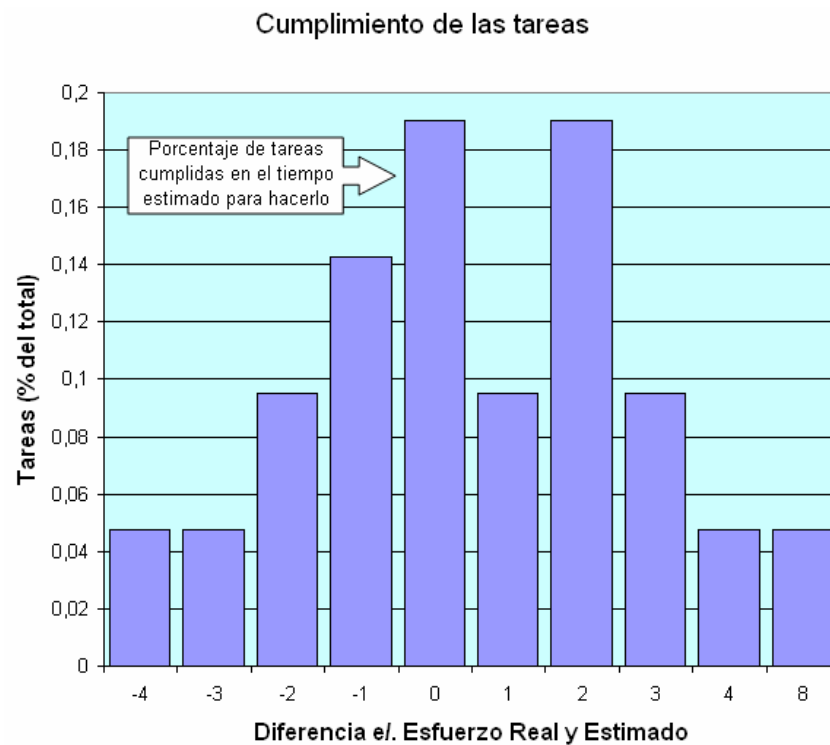


**Figura 4.16**

El valor mínimo y máximo del esfuerzo estimado de las tareas es 1 y 56 respectivamente, pero es en el intervalo de esfuerzo estimado entre 1 y 24 que se acumulan el 90% de todas las tareas registradas.

Para separar valores muy grandes que pudiesen distorsionar el cálculo del error de estimación promedio, se escogen las tareas con un esfuerzo estimado entre 1 y 24 horas x hombre (90% de las tareas) y con una diferencia entre 1 y 6 horas /hombre entre el esfuerzo real y el estimado (58% de las tareas con un esfuerzo real mayor al estimado). De esta forma se determina que el error promedio en la estimación de las tareas del proyecto B es: 62%

En el proyecto C, se registraron 21 tareas desde el 3 de Julio de 2006 hasta el 2 de Octubre del 2006 (período en el cual se aplicó el estudio), de estas tareas un 33% consumieron menos horas/hombre de las estimadas, mientras que el 19% consumieron exactamente las horas/hombre estimadas, y el restante 48% consumieron más horas/hombre de las que fueron estimadas al inicio de cada una. En la figura 4.17 se observan los datos mencionados anteriormente. Fácilmente se aprecia que la mayoría de tareas que excedieron el esfuerzo estimado (43%) están en el rango de 1 a 4 horas/hombre de diferencia.



**Figura 4.17**

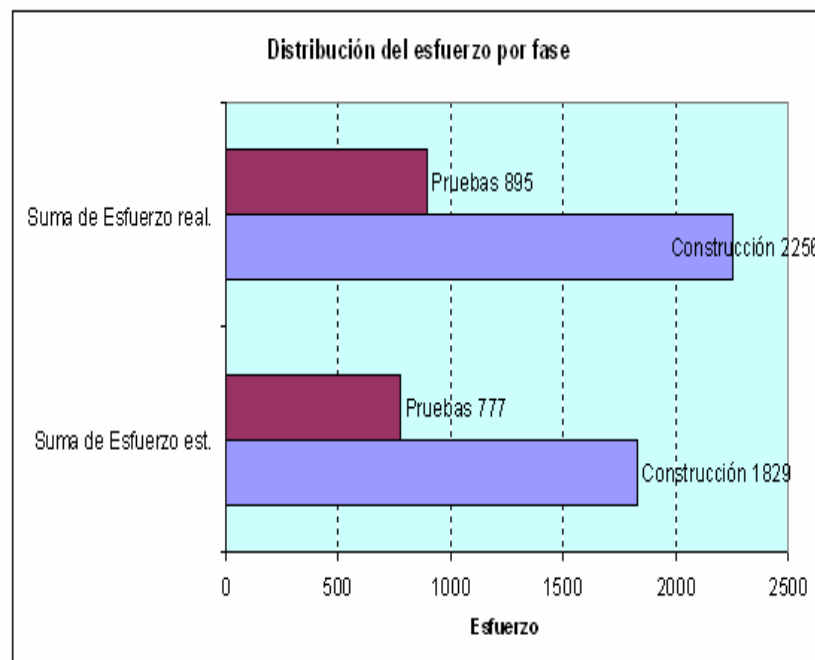
El valor mínimo y máximo del esfuerzo estimado de las tareas es 2 y 24 respectivamente, pero es en el intervalo de esfuerzo estimado entre 2 y 10 que se aglutina el 95% de todas las tareas registradas.

Para separar valores muy grandes que pudiesen distorsionar el cálculo del error de estimación promedio, se escogen las tareas con un esfuerzo estimado entre 2 y 10 horas x hombre (95% de las tareas) y con una diferencia entre 1 y 4 horas /hombre entre el esfuerzo real y el estimado (43% de las tareas con un esfuerzo real mayor al estimado).

De esta forma se determina que el error promedio en la estimación de las tareas del proyecto C es: 50%

#### 4.4.4. Distribución del esfuerzo por fase del proyecto

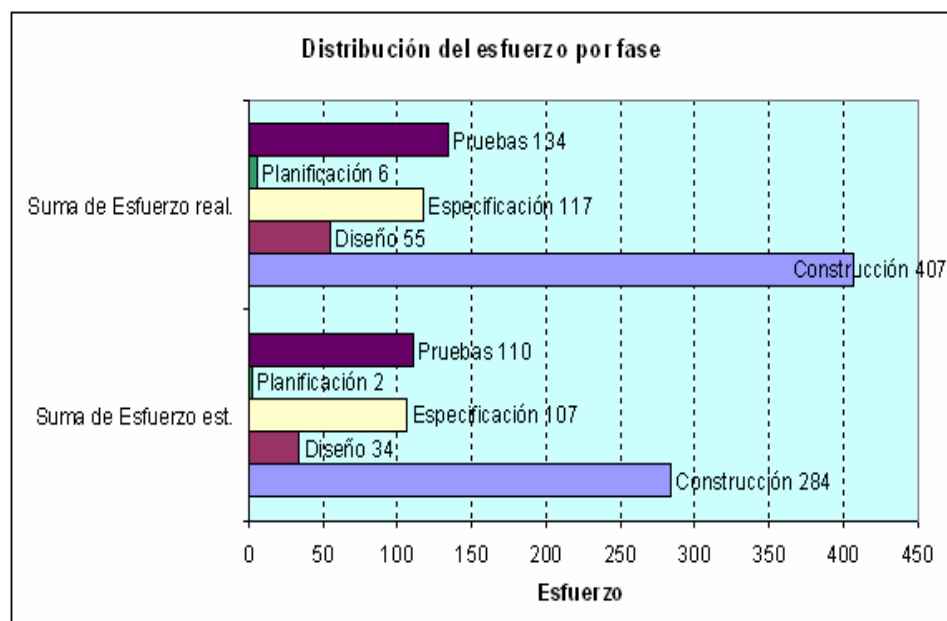
Durante el tiempo de recolección de datos del proyecto A, comprendido entre el 27 de Julio hasta el 13 de Octubre del 2006, el personal asignado al proyecto realizó actividades de Construcción y Pruebas, desglosadas así: 2256 horas-hombre en Construcción y 895 en Pruebas. En la figura 4.18 se puede notar que el esfuerzo real es mayor que el estimado en cada una de las fases.



**Figura 4.18**



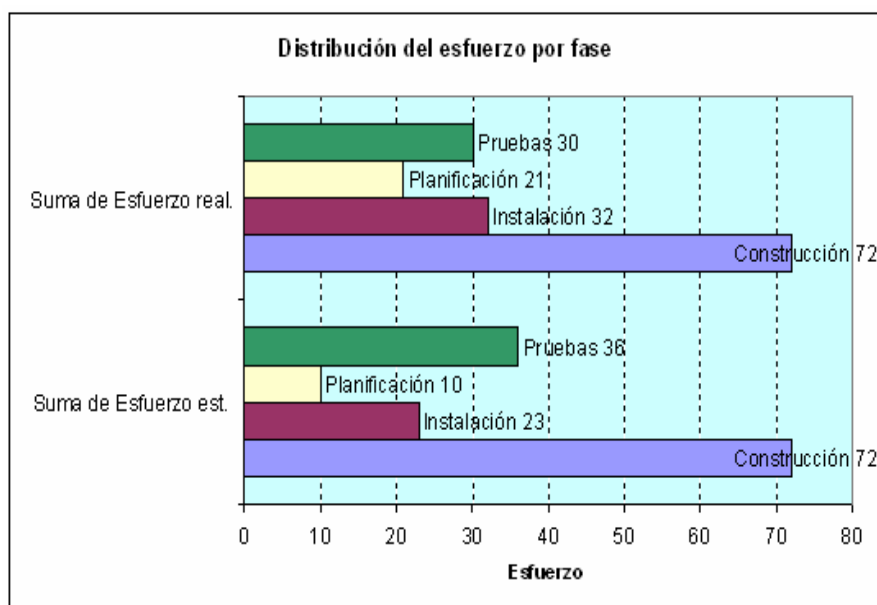
En el caso del proyecto B los datos obtenidos en horas-hombre fueron los siguientes: 6 en Planificación, 117 en Especificación, 55 en Diseño, 407 en Construcción y 134 en Pruebas. En la figura 4.19 se puede observar que el esfuerzo real es mayor al estimado en cada una de las fases.



**Figura 4.19**

Durante el tiempo de recolección de datos del proyecto C que comprendió el período entre el 3 de Julio de 2006 hasta el 2 de Octubre del 2006, el personal asignado al proyecto de software realizó actividades de Planificación, Construcción, Pruebas e Instalación siendo el esfuerzo empleado en horas-hombre el siguiente: 21 en Planificación, 72 en Construcción, 30 en Pruebas y 32 en Instalación.

En la figura 4.20 se puede apreciar que a diferencia de los proyectos A y B, éste sólo se excedió en la etapa de pruebas.



**Figura 4.20**

#### 4.4.5. Esfuerzo empleado en documentación

Para el proyecto A se obtuvo que el esfuerzo en documentación empleado representa el 3% (96 horas/hombre) del esfuerzo real registrado en desarrollo del proyecto (3151 horas/hombre), siendo las actas de reunión, y los reportes de actividades diarias los tipos de documentos que más se elaboraron.

Para el proyecto B, el esfuerzo en documentar representó el 1,3%, es decir 9,4 horas/hombre versus 719 horas/hombre del total del

proyecto, siendo los informes de actividades realizadas los únicos documentos que se elaboraron en este proyecto.

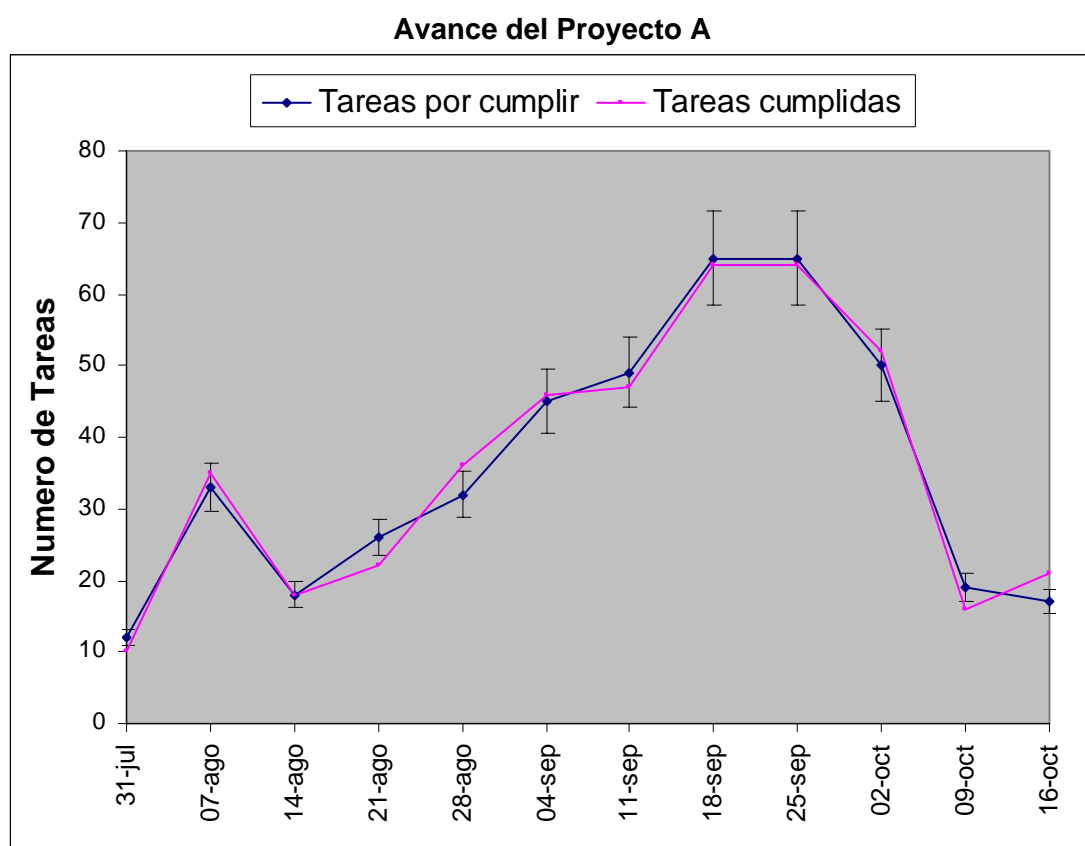
Durante el período de recolección de datos en el proyecto C no se elaboraron documentos, motivo por el cual el esfuerzo empleado para esta actividad fue cero.

#### **4.4.6. Avance del proyecto y complejidad**

El avance del proyecto se debía evaluar considerando la existencia de un cronograma de actividades elaborado al inicio del proyecto y las diferentes actividades realizadas durante el desarrollo del proyecto. Sin embargo, las empresas que participaron en el proyecto piloto sólo realizaban una planificación de las tareas conforme iban avanzando semana a semana, nunca hicieron un cronograma general al inicio del proyecto. Debido a esta forma de trabajar, observamos que el indicador de avance de proyecto presenta una información que no necesariamente refleja o se aproxima al avance real que tuvieron estos proyectos.

En la figura 4.21 cada punto de la línea azul representa el número de tareas que se planificaron, mientras que la línea rosada indica el número real de tareas cumplidas para las mismas fechas. Las líneas

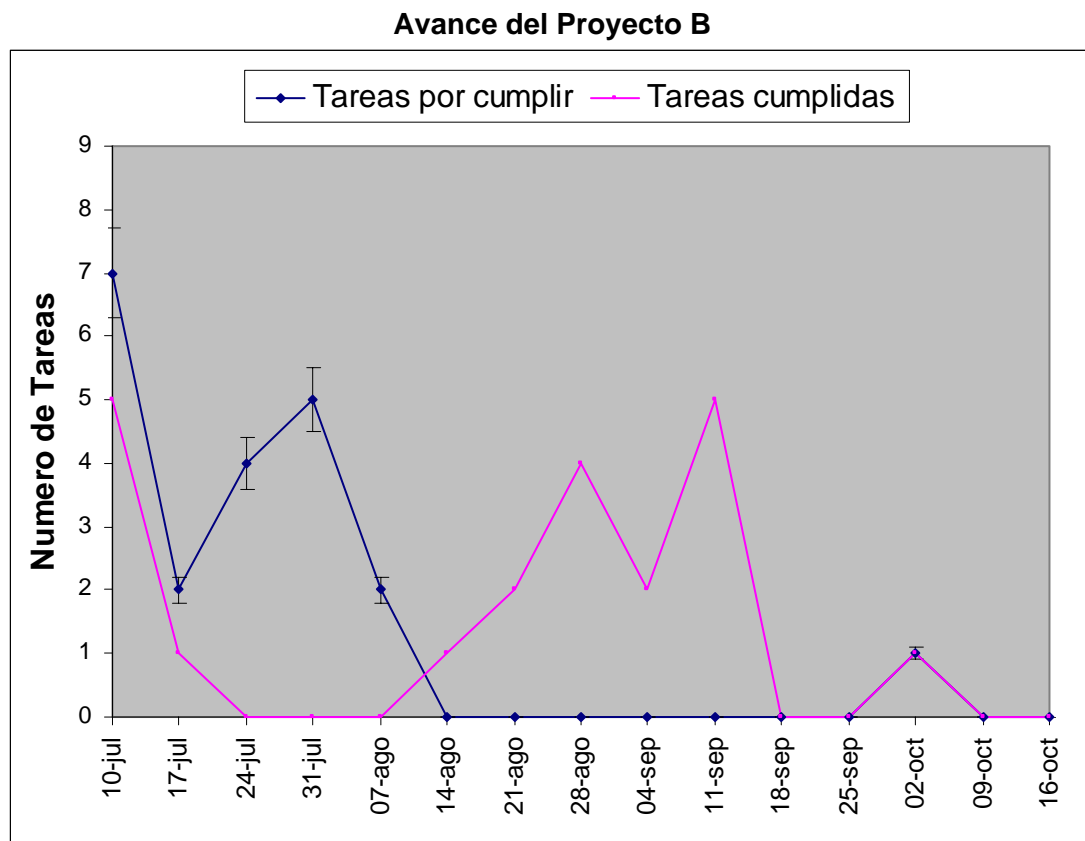
verticales sobre cada punto en la línea azul representan un porcentaje de error escogido (en este caso 10%) para evaluar la diferencia entre las tareas por cumplir y las cumplidas. En el caso del proyecto A, se aprecia que en 5 ocasiones (5 puntos de la línea rosa están fuera del margen de error) no alcanzaron a terminar las tareas planificadas.



**Figura 4.21**

El proyecto B obtuvo resultados similares al A. El proyecto C, difiere de los anteriores puesto que éste enfrentó prolongadas interrupciones durante su desarrollo que se pueden apreciar en la figura 4.22. Esto

se debió, fundamentalmente, a que el recurso asignado a este proyecto tenía a su cargo tareas de soporte al cliente, las mismas que se convirtieron en prioritarias, postergando el desarrollo del proyecto que estaba siendo monitoreado.



**Figura 4.22**

Para evaluar la **complejidad** se utilizó el método descrito en [24] y el instrumento de medición aplicado en un estudio anterior [34] a nuestra recopilación de datos.

Los valores de complejidad de los tres proyectos se pueden observar en la figura 4.23 en la que se destaca que los valores de complejidad de negocio son similares en los tres proyectos, pero con diferentes valores de complejidad técnica del proyecto ya que se trata de tres diferentes tipos de proyectos de desarrollo. Finalmente estos resultados muestran que los tres proyectos se clasifican como proyectos que enfrentan una situación de negocio difícil.

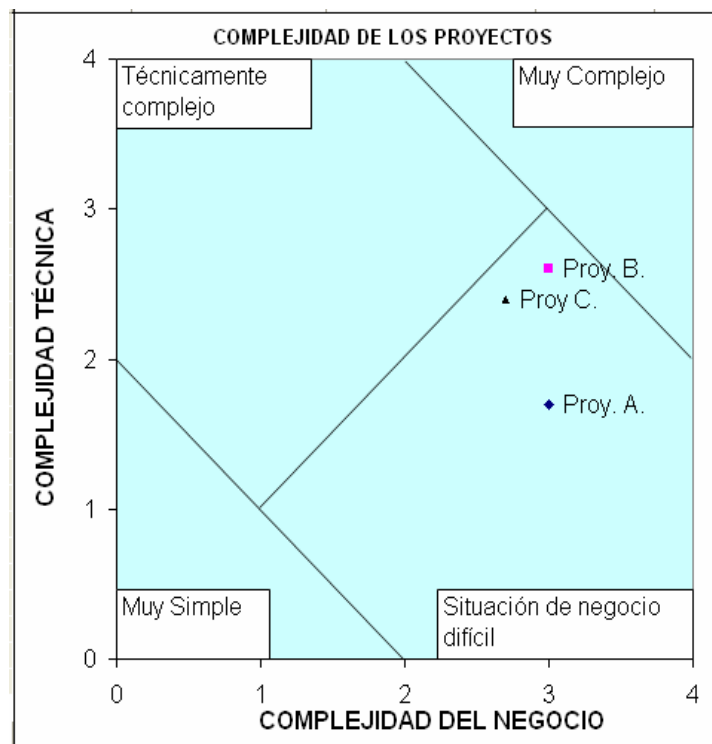


Figura 4.23

#### 4.4.7. Inspecciones durante el proceso de desarrollo

Durante el período de recolección de datos del proyecto A se registraron 59 inspecciones: 43 en la fase de Construcción y 16 en Pruebas. El esfuerzo empleado en estas inspecciones dió como resultado 412 horas/hombre consumiendo 319 horas/hombre en Construcción y 93 horas/hombre en Pruebas, esto representa un 13% del esfuerzo total consumido por el desarrollo del proyecto. Las actividades de inspección más realizadas fueron revisiones de cronogramas, reportes de avances, avances de módulos y control de cambios.

En el proyecto B se registraron 2 inspecciones: 1 en la fase de Diseño y 1 en Construcción. Estas inspecciones emplearon un esfuerzo real de 36 horas/hombre: 32 horas/hombre en Diseño y 4 en Construcción lo que significó un 5% del esfuerzo total consumido por el desarrollo del proyecto. La Tabla 4.3 resume los casos analizados.

Proyecto	Nro. de Inspecciones	Esfuerzo			
		Diseño	Construcción	Pruebas	% del Total
A	59	0	319	93	13,1%
B	2	32	4	0	5,0%

**Tabla 4.3: Inspecciones durante el proceso**

Durante el período de recolección de datos no se realizaron actividades de inspección en el proyecto C.

#### **4.4.8. Índice de rendimiento de tiempo y costo**

A través de los índices de rendimiento de tiempo y de costo, podemos determinar el grado de cumplimiento con el cronograma planificado y la tasa a la cual se consume el presupuesto del proyecto. Estos valores se calcularon para los proyectos A y B, mientras que para el C, esta información se mantuvo reservada.

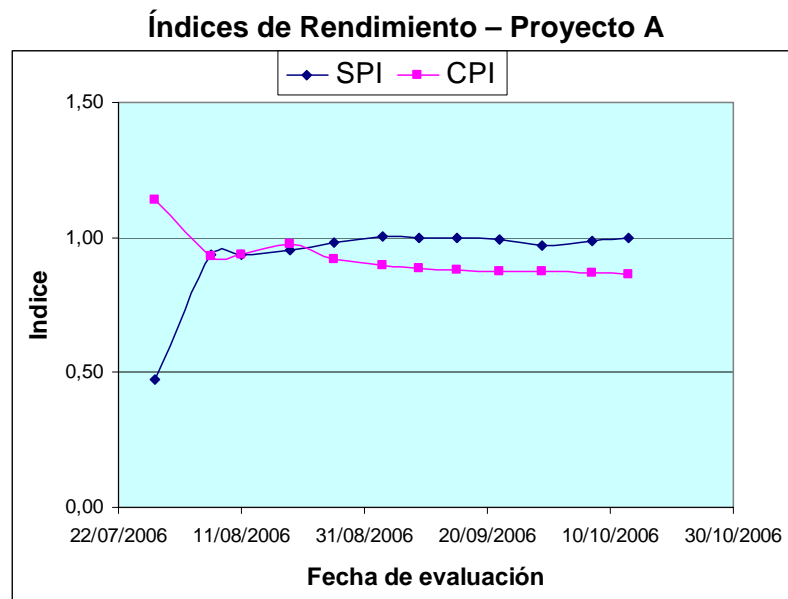
El índice CPI (índice de rendimiento de costo) se calcula al dividir el costo estimado del trabajo realizado para el costo real de este trabajo desarrollado. Mientras que el índice SPI (índice de rendimiento de tiempo) se calcula al dividir el costo estimado del trabajo realizado para el costo estimado del trabajo planificado a desarrollar desde la fecha de inicio hasta la fecha de evaluación de estos valores.

Para el caso del proyecto A la primera semana de evaluación presentó valores de CPI y SPI de 1,14 y 0,48 respectivamente. Estos valores indican que las tareas realizadas hasta esa fecha 28/07/2006 se las cumplió a un costo menor al estimado pero por detrás del tiempo planificado para cumplirlas. Sin embargo consideramos que estos



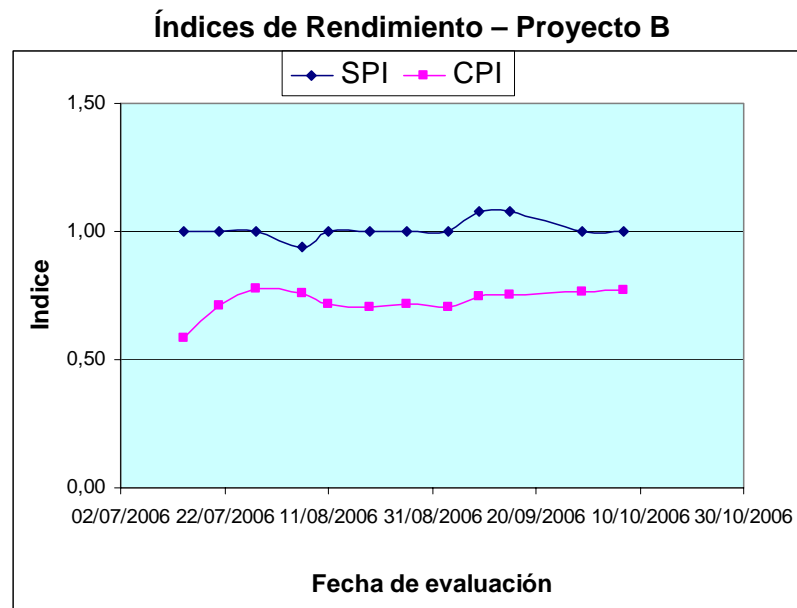
valores se dieron de esta forma porque recién se empezó la actividad de registro de las tareas. La realidad se podría observar en los siguientes puntos de evaluación del proyecto.

De forma general al observar los valores de la figura 4.24 se puede decir que las diferentes tareas de este proyecto se cumplieron de acuerdo al tiempo estimado ya que los diferentes valores de SPI desde el 19/08/2006 hasta el 13/10/2006 están sobre el valor 1 del eje vertical. Sin embargo el valor CPI se ha mantenido por debajo del valor 1 del eje vertical con tendencia a ir disminuyendo a medida que avanza el tiempo, lo que nos indica que el costo real de las tareas que se realizaron han ido en constante aumento respecto de su valor estimado.



**Figura 4.24**

Al analizar el caso del proyecto B observando los valores de la figura 4.25 se puede decir que las tareas de este proyecto se cumplieron de acuerdo al tiempo estimado ya que los diferentes valores de SPI desde el 14/07/2006 hasta el 07/10/2006 están sobre el valor 1 del eje vertical a excepción de 3 registros. Se puede notar también que los valores de CPI se han mantenido por debajo del valor 1 del eje vertical (tomando valores entre 0,78 y 0,71 con excepción de la primera semana) con tendencia a mantenerse a medida que avanza el tiempo, lo que nos indica que el costo real de las tareas que se realizaron es en promedio un 25% mayor al estimado.



**Figura 4.25**

#### 4.4.9. Tasa de variación en los requerimientos del software

Debido a que los proyectos ya habían iniciado cuando empezamos a monitorearlos, nos vimos obligados a consultar documentos que poseían las empresas para contabilizar requerimientos iniciales y los cambios que se dieron antes de iniciar este estudio.

En la fase de Construcción es donde se presentaron la mayor cantidad de cambios de requerimientos. La Tabla 4.4 muestra los requerimientos iniciales de los proyectos y la tasa de variación de requerimientos agregados, modificados y eliminados.

Proyecto	Requerimientos			
	Iniciales	Agregados	Modificados	Eliminados
A	14	42%	278%	0
B	23	21%	73%	0
C	14	0	14%	0

**Tabla 4.4 Cambios de Requerimientos**

Durante el tiempo de recolección de datos del proyecto A se registraron 45 cambios de requerimientos, 32 fueron cambios Funcionales mientras que 13 fueron No Funcionales. En el proyecto B se registraron 3 cambios de requerimientos, 1 Funcional y 2 No Funcionales. En el proyecto C no se registraron cambios de requerimientos, motivo por el cual no se pudo realizar el análisis para este caso.

#### **4.4.10. Defectos y fallas**

En nuestra tesis el término “**defecto**” quiere decir algún tipo de problema que se presentó durante el desarrollo del proyecto causando: mal funcionamiento de la aplicación de software, retrabajo y/o demoras en la entrega del producto. El término “**falla**” hace mención a interrupciones inesperadas en el funcionamiento de la aplicación de software cuando está siendo utilizada por el cliente.

Anteriormente se mencionó que los proyectos que se evaluaron ya habían iniciado el proceso de desarrollo y su finalización estaba planificada para tiempo después del período de nuestra evaluación, por lo que en principio se podría pensar que por su definición no habría un registro de “fallas” ya que el producto no estaría terminado para hacer una entrega formal al cliente. Sin embargo, del proyecto A pudimos registrar las fallas debido a que cada una de sus partes fue puesta en producción inmediatamente después de pasar la etapa de pruebas.

Los valores correspondientes a los defectos y fallas, y su distribución en el tiempo se presentan en los siguientes apartados.

#### **4.4.11. Fuga de defectos**

Para el proyecto A se registraron un total de 13 Fallas y 52 defectos, si calculamos la tasa de fuga de defectos con la fórmula (fuga de defectos = fallas / (fallas+defectos)) obtenemos una tasa 0,2 que nos indica que el 20% de los problemas que se han registrado corresponden a “fallas” que son problemas que el cliente ha detectado y reportado de los componentes que se le entregaron.

Para los proyectos B y C no se pudo realizar este análisis debido a que estos no entregaron partes del software al cliente antes de terminar con sus fases de Construcción y Pruebas, las cuales todavía no concluían hasta la última fecha de recolección de datos para el presente estudio.

#### **4.4.12. Distribución por fase del proyecto**

En los tres proyectos evaluados se registraron defectos durante las fases de Construcción y Pruebas.

Para el caso del proyecto A se registraron 52 defectos y 13 fallas distribuidos de la siguiente manera: en la fase de Construcción se presentó el 59% de defectos y el 8% de fallas, mientras que en la fase de Pruebas se presentó el 41% de defectos. También se analizó la severidad de estos problemas y se obtuvo que en la fase de Construcción el 3% fueron críticos, el 22% graves y el 75% leves; en la fase de Pruebas el 10% fueron críticos, el 43% graves y el 47% leves.

En el caso del proyecto B sólo se registraron 3 defectos, 2 en la fase de construcción (un grave y un leve) y 1 en la de pruebas (leve).

Para el caso del proyecto C sólo se registraron 2 defectos, 1 en la fase de Construcción con severidad grave y otro en la fase de Pruebas con severidad leve.

#### **4.4.13. Distribución por tiempo transcurrido del proyecto**

Se analizó la distribución de defectos mientras transcurrían las semanas del período de recolección de datos, durante este período para el caso del proyecto A se obtuvo una cantidad de datos considerable para realizar el análisis en el cual se pudo observar como iba aumentando la cantidad de defectos y fallas a medida que avanzaba el desarrollo del proyecto, en la cuarta semana de evaluación se presentaron la mayoría de problemas y después comenzó a disminuir la cantidad de los mismos. Las figuras 4.26 y 4.27 resumen la distribución de los defectos y fallas respectivamente durante el período de recolección de datos del proyecto A.

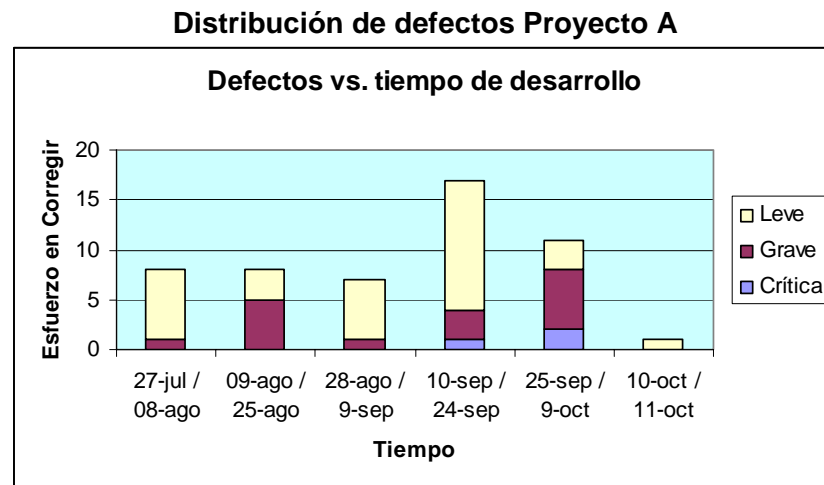


Figura 4.26

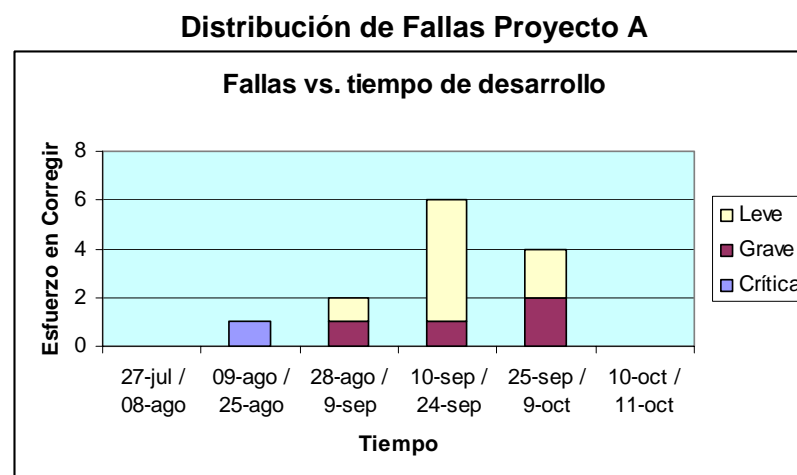


Figura 4.27

Para el caso del proyecto B y el proyecto C no se obtuvieron una cantidad de datos suficientes para hacer un resumen similar.



#### 4.4.14. Causa – Efecto

Un registro de la cantidad de defectos, su clasificación por severidad y el esfuerzo que se emplea en corregirlos no representan por si solos un aporte a la mejora de los procesos. Debemos de conocer las causas que provocan estos problemas, y para ello utilizamos el diagrama causa-efecto. Los tres proyectos enfrentaron problemas similares por lo que presentamos un solo diagrama causa-efecto, el mismo que se generó de la revisión de todos los problemas que fueron reportados. Escogimos como efecto los retrasos en el desarrollo del software.

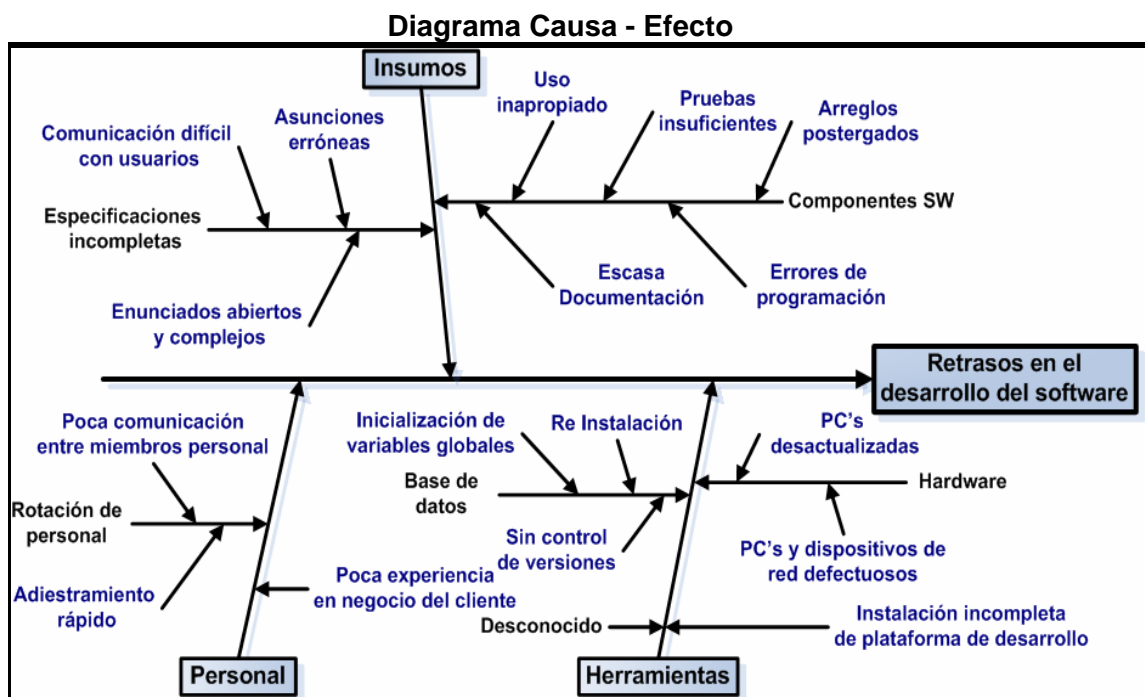


Figura 4.28

Las categorías de un diagrama Causa-Efecto son: Personal, Insumos, Métodos y Herramientas. El gráfico de arriba no posee la categoría “Métodos”, esto se debe a que en ninguno de los reportes de problemas mencionaron la falta de métodos, estándares o procedimientos debidamente establecidos. Esto no significa que no existan quejas por esta situación, sino que hace explícito el desconocimiento que tiene el personal de las tres empresas de que la falta de procedimientos y estándares es una de las mayores causas de los problemas a los que se enfrentan. Esto último es “nuestro criterio personal”.

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones acerca del estudio exploratorio

1. La realización de estudios dirigidos a obtener información que nos ayudan a mejorar nuestro conocimiento acerca del proceso de desarrollo del software se ven facilitados cuando en las empresas de software existe un seguimiento cuantitativo y cualitativo de factores clave como tiempo y metodologías de desarrollo, el costo y esfuerzo del proyecto de software y la calidad del producto desarrollado.
2. El uso de métricas en nuestras empresas no es común y depende mucho del avance del proyecto, ya que al experimentar retrasos la actividad de recopilación de datos para la formación de métricas se suspende, debido principalmente a que la documentación del proyecto se posterga.
3. Se determinó que actualmente la codificación de la aplicación de software es la actividad de mayor duración, costo y consumo de recursos durante el proceso de desarrollo de software. Siendo esta una situación que se debe buscar revertir especialmente poniendo mayor atención a la especificación de requerimientos.

4. Los costos del mantenimiento del software se incrementan debido principalmente a la rotación del personal, la falta de documentación y un número insuficiente de pruebas. Esto ocasiona que el software entregado al cliente posea defectos aunque en un porcentaje mínimo corrigiéndolos al ser reportados y en la mayoría de los casos sin lamentar impactos negativos sobre los datos de los clientes.

### **Conclusiones acerca de la aplicación de las métricas en las tres empresas de software.**

1. La redacción de los objetivos de medición que se llegaron a definir reflejan el grado de madurez que tienen las empresas en la aplicación y uso de métricas, pues los términos como: *conocer*, *determinar*, y *monitorear* (en este caso, el proceso de desarrollo de software), son los primeros que se utilizan antes de pasar a términos más “desafiantes” como: *mejorar*, *disminuir*, o *sostener* que implican un conocimiento previo y cuantitativo del estado actual de las prácticas productivas utilizadas en cada empresa.
2. El grado de satisfacción del cliente se ve positivamente afectado con una mayor participación del cliente y los usuarios en el ciclo de desarrollo del proyecto de software.

3. En los tres proyectos evaluados coincide que la fase que consume la mayor cantidad del esfuerzo empleado es la fase de construcción (72% en el proyecto A, 57% en el proyecto B y 46% en el proyecto C). Coincidente con lo mencionado el rol más desempeñado por el personal de la empresa es programador o desarrollador, en contraste a los altos porcentajes empleados en la construcción del software, la documentación de los proyectos es escasa (3% en el proyecto A, 1.3% en el proyecto B y 0% en el proyecto C) y limitada a las actas de reuniones, reporte de actividades y revisión de cronogramas. Finalmente se determinó un error promedio en la estimación del esfuerzo requerido en los tres proyectos evaluados fue del 50%.
  
4. Los valores de “complejidad de los negocios” en promedio fueron de 2.9 lo que ubica a estos proyectos en la categoría de situación de negocio difícil, razón por la cual consideramos que es una de las causas principales de los retrasos que han experimentado los tres proyectos evaluados.
  
5. El esfuerzo empleado para realizar inspecciones fue escaso en los tres proyectos (13% en el proyecto A, 5% en el proyecto B y 0% en el proyecto C).

6. Los índices de rendimiento de tiempo y costo reflejaron atrasos en el cronograma y un gasto mayor al estimado respectivamente aunque advertimos que no son del todo exactos por cuanto no hubo una planificación inicial de los proyectos, sino que esta se lo hacía conforme avanzaba el proyecto.
7. Se agregaron y se modificaron requerimientos en los tres proyectos evaluados sin embargo no se registro la eliminación de ningún requerimiento.
8. Se determinó que el 80% de todos los problemas registrados corresponden a las siguientes categorías: 1) Errores de programación (44%), 2) Problemas con la reutilización de componentes (19%), 3) Funcionalidad incompleta (11%) y 4) Desconocimiento de las herramientas de software (6%). Siendo el efecto principal que se generó los retrasos en el desarrollo del proyecto.
9. Finalmente, las métricas, en esta aplicación inicial, no mejoraron la actual forma de “hacer las cosas” al interior de la empresa, pero si ayudaron a conocer el estado y capacidades actuales de las prácticas que se utilizan.

## Recomendaciones

1. Para el seguimiento y control del avance del proyecto de software es aconsejable utilizar técnicas de seguimiento y control de proyectos descritas en la literatura (técnica nominal de grupos, análisis de valor devengado, encuestas, etc.) pero adaptadas a la realidad y prácticas actuales de la empresa. Debido principalmente a que existen técnicas más “sofisticadas” como el análisis estadístico de procesos, modelos de predicción, entre otros que requieren un entorno (La cultura organizacional de la empresa, capacidades del personal e infraestructura que de soporte a esta técnicas de seguimiento y control de proyectos) que permita su adecuada implementación.
2. Para disminuir la resistencia al cambio es mejor desarrollar herramientas que complementen o hagan uso de la actual forma de trabajar de las empresas, al menos como primer paso para introducirlos en la mejora de procesos. Esto quiere decir ser sencillos y concretos en la elaboración de instrumentos de medición y recopilación de datos.
3. A pesar de que en la bibliografía sobre métricas se recomienda el conteo de líneas de código, esta métrica, no es viable de aplicar en nuestro medio por las siguientes razones: 1) Se las considera de poco valor

práctico. 2) En una misma empresa se usan diferentes lenguajes de programación lo que dificulta establecer comparaciones entre proyectos, y 3) No se usan estándares para la codificación que ayuden a desarrollar un código consistente y fácil de seguir.





# Apéndices

Escuela Superior Politécnica del Litoral  
Facultad de Ingeniería en Electricidad y Computación  
Componente 8 – VLIR

Aspectos de la calidad y dificultades durante la administración de proyectos  
"Estudio exploratorio"

**Introducción:** El presente cuestionario esta dirigido a la(s) persona(s) que en una empresa de desarrollo de software tengan a cargo actividades como: la estimación del proyecto, el seguimiento de los recursos, la planificación, etc. Estas personas pueden ser el administrador de proyectos, supervisor de equipos de desarrollo, incluso el mismo gerente de la empresa. Con el propósito conocer aspectos de la administración de los proyectos de software Las respuestas son anónimas, y de antemano le agradecemos su sinceridad y colaboración.

1. ACERCA DEL ENTREVISTADO

- 1.1 Género: A) Masculino  B) Femenino
- 1.2 Edad: \_\_\_\_\_
- 1.3 Máximo nivel de instrucción: A) Superior  B) Postgrado
- 1.4 Años en la compañía: \_\_\_\_\_
- 1.5 Cargo que desempeña en la organización: (Marque con una X sólo una opción)
- |                                |                          |            |                          |
|--------------------------------|--------------------------|------------|--------------------------|
| Gerente                        | <input type="checkbox"/> | Supervisor | <input type="checkbox"/> |
| Administrador/ Líder de equipo | <input type="checkbox"/> | Otro       | <input type="checkbox"/> |
- Indique: \_\_\_\_\_

2. ACERCA DE LA EMPRESA

- 2.1 El número de empleados de la empresa es: \_\_\_\_\_
- 2.2 La empresa tiene certificación ISO 9001 2000: A) Si  B) No
- 2.3 La empresa tiene \_\_\_\_\_ años de funcionamiento
- 2.4 Respecto del software para exportación indique:
- |   | Si                       | No                       |
|---|--------------------------|--------------------------|
| A) Actualmente exportan software  | <input type="checkbox"/> | <input type="checkbox"/> |
| B) En el pasado exportaron software   | <input type="checkbox"/> | <input type="checkbox"/> |
| C) <u>Si No exportan</u> , esperan exportar software en el corto plazo ( menos de un año) | <input type="checkbox"/> | <input type="checkbox"/> |
| D) <u>Si No exportan</u> , esperan exportar software en el largo plazo (mas de un año)    | <input type="checkbox"/> | <input type="checkbox"/> |

Si contesto **Si** a los literales a) o b), pase a la pregunta 2.5  
Si contesto **No**, continúe con la siguiente sección.

- 2.5 Del total del software producido por la empresa, qué porcentaje representa el exportable en términos de:
- |                          | Actualmente | En el pasado |
|--------------------------|-------------|--------------|
| A) Valor Facturado       | ____%       | ____%        |
| B) Software Desarrollado | ____%       | ____%        |

### 3. GENERALIDADES DEL DESARROLLO DEL SOFTWARE

3.1 Con qué frecuencia documentan el proceso de desarrollo del software?

(Marque con una **X** sólo una opción)

Siempre  Casi Siempre  Regularmente  Pocas veces  Nunca

3.2 Señale los motivos que dificultan documentar el proceso de desarrollo del software

(Señale con una **X** todas las que aplican)

- A) Premura por terminar el proyecto   
 B) Desconocimiento de cómo documentar   
 C) Poco interés en realizar esta actividad   
 D) Se desconocen los beneficios de documentar   
 E) Otro

Si marcó el literal E), por favor indique:

3.3 El modelamiento(s) de procesos de software utilizado(s) por la empresa según el tipo de proyecto es:

(Marque con una **X** todas las que aplican)

Proyectos:

	*Pequeños	**Medianos	***Grandes	
A) Cascada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
B) Incremental	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	* Tiempo para culminación entre 1 y 6 meses
C) Espiral	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	** Tiempo para culminación entre 7 y 15 meses
D) RUP*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	***Tiempo para la culminación más de 15 meses
E) Time boxing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
F) Programación extrema	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
G) Otro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Si marcó el literal G), por favor indique:

Pequeño \_\_\_\_\_  
 Mediano \_\_\_\_\_  
 Grande \_\_\_\_\_

**Cascada** – Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas y consecutivas del proceso, como especificación de requerimientos, diseño de software, implementación, pruebas, etc. Nunca puede retornar a la etapa anterior (Ian Sommerville).

**Incremental** – En el desarrollo incremental, los clientes identifican de manera somera los servicios que provee el sistema. Identifican cuales son mas importantes y menos importantes se definen varios incrementos en donde cada uno proporciona un subconjunto de funcionalidad del sistema (Ian Sommerville).

**Espiral** – Este modelo de proceso del software se representa como una espiral en la que cada ciclo se divide en 4 partes (1. Definición de objetivos, 2. Evaluación y reducción de riesgos, 3. Desarrollo y validación, 4. Planeación) La diferencia importante entre este modelo y los otros modelos de proceso del software es la consideración explícita del riesgo. La construcción de prototipos se utiliza para resolver las dudas en los requerimientos y así reducir el riesgo (Ian Sommerville).

**\*Rational Unified Process** -El RUP integra las mejores prácticas de desarrollo de software a través de la definición de procesos, flujos de actividades, roles, guías, documentos patrón, ejemplos y métricas, con esto se obtiene un dramático ahorro de tiempo y esfuerzo en la implantación.

**Time boxing** – Modelo de desarrollo de software que asigna un determinado número de horas para completar una tarea y al final de dicho periodo la tarea debe estar 100% terminada. No se permiten retrasos. Time boxing es usado para definir la meta de la tarea, la estrategia para llevarla a cabo, la forma en que se verificarán los resultados y como se estará seguro de que cuando la tarea esta terminada esté “realmente terminada”.

**Programación extrema** – Es una metodología de administración y desarrollo de proyectos, cuyas características incluyen: La importancia de la comunicación constante dentro de la organización de desarrollo y con el cliente, la sencillez radical y la programación por pares.

3.4 Por favor, indique el número de personas designadas para el desarrollo de un:



Los datos aquí proporcionados son de carácter **confidencial** y los resultados corresponderán a la población analizada y no a empresas por separado.



(Indique **solo los que aplican**)

- A) Proyecto pequeño \_\_\_\_\_  
 B) Proyecto mediano \_\_\_\_\_  
 C) Proyecto grande \_\_\_\_\_

**3.5** Seleccione de entre las alternativas mostradas cuáles corresponden a metas de su empresa, en cuanto al desarrollo de productos de software, y asigne un número de acuerdo a su prioridad (Para la más importante 1, la siguiente 2 y así sucesivamente).

(Escoja **solo las que aplican** en su empresa)

- A) Maximizar Satisfacción del cliente. \_\_\_\_\_  
 B) Minimizar el esfuerzo de ingeniería \_\_\_\_\_  
 C) Minimizar los defectos \_\_\_\_\_  
 D) Minimizar la distribución del tiempo \_\_\_\_\_  
 E) Mejorar la productividad \_\_\_\_\_  
 F) Minimizar costos \_\_\_\_\_  
 G) Mejorar la calidad del software \_\_\_\_\_  
 H) Otro \_\_\_\_\_

Si marcó el literal H), por favor indique:

.....

**3.6** Por favor, indique la distribución del TIEMPO consumido (*en porcentaje*) en el desarrollo de proyectos de software.

(Complete **solo las filas que aplican**)

Distribución de tiempo en:

	Especificación / Requerimientos	Diseño	Codificación / Prueba de unidades	Integración / Pruebas del sistema	Desconoce los porcentajes
Pequeños	____[%]	____[%]	____[%]	____[%]	<input type="checkbox"/>
Medianos	____[%]	____[%]	____[%]	____[%]	<input type="checkbox"/>
Grandes	____[%]	____[%]	____[%]	____[%]	<input type="checkbox"/>

\* Se excluye la actividad de “operación y mantenimiento”, pues en esta actividad el software esta terminado y se instala para su uso práctico.

**3.7** Por favor, indique la distribución del COSTO consumido (*en porcentaje*) en el desarrollo de proyectos de software.

(Complete **solo las filas que aplican**)

Distribución de tiempo en:

	Especificación / Requerimientos	Diseño	Codificación / Prueba de unidades	Integración / Pruebas del sistema	Desconoce los porcentajes
Pequeños	____[%]	____[%]	____[%]	____[%]	<input type="checkbox"/>
Medianos	____[%]	____[%]	____[%]	____[%]	<input type="checkbox"/>
Grandes	____[%]	____[%]	____[%]	____[%]	<input type="checkbox"/>

3.8 Señale la fase del desarrollo del software que normalmente consume más recursos según el tamaño del proyecto.

(Marque con una X solo una opción por fila)

		Ciclo de vida del software				
		Especificación / Requerimientos	Diseño	Codificación / Prueba de unidades	Integración / Pruebas del sistema	Desconoce
Proyectos	Pequeños	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Medianos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Grandes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3.9 Conoce el concepto de "métrica del software"?

A) Si  B) No

3.10Cuál(es) de los siguientes términos le son familiares?

(Señale con una X todas las que aplican)

- |   |                          |                         |                          |
|---|--------------------------|-------------------------|--------------------------|
| A) Atributos del producto FURPS*          | <input type="checkbox"/> | D) Densidad de defectos | <input type="checkbox"/> |
| B) Complejidad ciclomática                | <input type="checkbox"/> | E) Puntos de función    | <input type="checkbox"/> |
| C) Despliegue de función de calidad QFD** | <input type="checkbox"/> | F) ROI***               | <input type="checkbox"/> |

\* Functionality Usability Reliability Performance Supportability

\*\* Quality Function Deployment

\*\*\*Return on Investment

3.11 Realizan mediciones del proceso de desarrollo de software en la empresa?

A) Si  B) No

3.12 Qué mediciones realizan en la empresa?

(Indique las mediciones que realiza y registra)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3.13 Señale si miden el tamaño del sistema en términos de:

(Señale con una X todas las que aplican)

- A) Líneas de código fuente
- B) Puntos de función
- C) Otro

Si marcó el literal C), por favor indique:

.....

**3.14** Indique cuáles son los aspectos del desarrollo del software que ACTUALMENTE mide, y/o los que consideraría necesarios medir en el CORTO PLAZO (menos de 1 año).

*(Señale con una X todas las que apliquen)*

		Actualmente	En el corto plazo
A)	Tamaño del sistema	<input type="checkbox"/>	<input type="checkbox"/>
B)	Complejidad de los módulos	<input type="checkbox"/>	<input type="checkbox"/>
C)	Progreso del desarrollo	<input type="checkbox"/>	<input type="checkbox"/>
D)	Re-utilización de código	<input type="checkbox"/>	<input type="checkbox"/>
E)	Recursos utilizados	<input type="checkbox"/>	<input type="checkbox"/>
F)	Costos	<input type="checkbox"/>	<input type="checkbox"/>
G)	Defectos	<input type="checkbox"/>	<input type="checkbox"/>
H)	Actividades de mantenimiento	<input type="checkbox"/>	<input type="checkbox"/>
I)	Estabilidad de los requerimientos	<input type="checkbox"/>	<input type="checkbox"/>
J)	Calidad del software	<input type="checkbox"/>	<input type="checkbox"/>
K)	Resultados de las pruebas	<input type="checkbox"/>	<input type="checkbox"/>
L)	Documentación generada	<input type="checkbox"/>	<input type="checkbox"/>

**3.15** En su empresa, llevan un **REGISTRO del tiempo en horas** que ocupan en realizar las diferentes actividades que demanda el proceso de desarrollo de software?

A) Si  B) No

Si contesto **Si**, continúe con la siguiente pregunta.

Si contesto **No**, salte a la pregunta 3.17

**3.16** Indique en cual de las siguientes actividades llevan un control de las horas consumidas:

*(Marque con una X todas las que aplican)*

- A) Recopilar requerimientos
- B) Planear
- C) Diseñar
- D) Reuniones de trabajo
- E) Codificar
- F) Probar el sistema
- G) Instalar el sistema
- H) Corregir errores
- I) Otro

Si marcó el literal I), por favor indique:

.....

.....

.....

3.17 Cuál de las actividades de desarrollo del software es la principal fuente de defectos, según el tamaño del proyecto?

(Marque con una X solo una opción)

		Especificación / Requerimientos	Diseño	Codificación / Prueba de unidades	Integración / Pruebas del sistema
Proyectos	Pequeños	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Medianos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Grandes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Si contesto Si, a la pregunta 3.11, continúe con las siguientes preguntas.

Si contesto No, pase a la siguiente sección.

3.18 Con qué frecuencia los resultados de las mediciones se utilizan para la toma de decisiones?

(Marque con una X solo una opción)

Siempre  Casi Siempre  Regularmente  Pocas veces  Nunca

3.19 De qué forma recopilan los datos necesarios para las mediciones?

(Marque con una X todas las que aplican)

- A) Automáticamente, (Con la ayuda de programas computacionales)
- B) Manualmente, (Usando la documentación del proyecto)

#### 4. ESPECIFICACION / REQUERIMIENTOS

4.1 Por favor, indique si antes del desarrollo del software realiza la(s) estimación(es) de:

(Señale con una X todas las que aplican)

- |                       |                          |                  |                          |
|-----------------------|--------------------------|------------------|--------------------------|
| A) Tamaño del sistema | <input type="checkbox"/> | E) Documentación | <input type="checkbox"/> |
| B) Recursos           | <input type="checkbox"/> | F) Tiempo        | <input type="checkbox"/> |
| C) Complejidad        | <input type="checkbox"/> | G) Otro          | <input type="checkbox"/> |
| D) Costos             | <input type="checkbox"/> |                  |                          |

Si marcó el literal G), por favor indique:

.....

4.2 Escoja y priorice los problemas mas comunes al momento de recibir los requerimientos del usuario, (Para el más común 1, el siguiente 2 y así sucesivamente).

(Seleccione solo los que aplican)

- |   |       |
|---|-------|
| A) El cliente desconoce el alcance (limites) del proyecto | _____ |
| B) Los requerimientos son confusos e inconsistentes*      | _____ |
| C) No se mencionan restricciones                          | _____ |
| D) Los requerimientos están incompletos                   | _____ |
| E) Los requerimientos están agrupados**                   | _____ |
| F) Otro   | _____ |

Si marcó el literal F), por favor indique:

.....

\* Varias personas de un mismo o de varios departamentos tienen criterios diferentes con respecto a un mismo requerimiento.

\*\* No están consientes de que un requerimiento abarca varios dentro de el.

**4.3 Utiliza datos históricos\* para hacer estimaciones?**

*(Marque con una X sólo una opción)*

Siempre  Casi Siempre  Regularmente  Pocas veces  Nunca

**Datos históricos\*** Datos recopilados y organizados de proyectos pasados sobre factores como el tiempo, los recursos, el tamaño, etc. que se utilizaron para llevarlos a cabo.

**4.4 Qué herramientas utiliza para planificar un proyecto?**

*(Señale con una X todas las que aplican)*

- A) Microsoft Project   
 B) Microsoft Excel   
 C) Microsoft Visio   
 D) Software propio   
 E) Otro

Si marcó el literal E), por favor indique:

.....

## 5. DISEÑO

**5.1 Con qué frecuencia realizan inspecciones en la fase de diseño?**

*(Marque con una X sólo una opción)*

Siempre  Casi Siempre  Regularmente  Pocas veces  Nunca

Si marcó **Nunca**, continúe con la pregunta 5.3

De lo contrario, continúe con la pregunta 5.2

**5.2 Indique el porcentaje de defectos de diseño que detectan con inspecciones.**

*(Marque con una X sólo una opción)*

[0-25%)  [25-50%)  [50-75%)  [75%-100%)  Desconoce

**5.3 Indique el tipo de diseño que utiliza la empresa.**

*(Señale con una X todas las que aplican)*

- A) Diseño estructurado  C) Patrones de Diseño   
 B) Orientado a objetos  D) Otros

Si marcó el literal D), por favor indique:

.....

**5.4 Al terminar la fase de diseño realiza una estimación más precisa del proyecto en relación a:**

*(Señale con una X todas las que aplican)*

- A) Tamaño del sistema  D) Recursos   
 B) Costos  E) Complejidad   
 C) Tiempo

## 6. CODIFICACIÓN, INTEGRACION Y PRUEBAS

**6.1 Re-utilizan componentes de software?**

A) Si  B) No

Si contesto **Si**, continúe con la siguiente pregunta.

Si contesto **No**, salte a la pregunta 6.5



6.2 De qué forma re-utilizan los componentes de software?

-----  
 -----  
 -----  
 -----

6.3 Indique en que porcentaje re-utilizan componentes de software.

*(Marque con una X solo una opción)*

[0-25%)  [25-50%)  [50-75%)  [75%-100%)  Desconoce

6.4 El tiempo y recursos que consume un proyecto que re-utiliza componentes de software respecto del tiempo y recursos necesarios para un proyecto nuevo son?

*(Marque con una X solo una opción)*

- A) Los mismos   
 B) La mitad   
 C) La cuarta parte   
 D) Otro

Si marcó el literal D), por favor indique:

-----

6.5 Para procurar controlar la cantidad y gravedad de los defectos se hacen?

*(Señale con una X todas las que aplican)*

- A) Inspecciones tempranas  D) Se controlan cuando estos se producen   
 B) Análisis de defectos  E) Otros   
 C) Revisiones periódicas

Si marcó el literal E), por favor indique:

-----

6.6 Qué personas están a cargo de probar el software?

*(Señale con una X todas las que aplican)*

- A) El mismo equipo de desarrollo  D) El cliente   
 B) Un equipo de desarrollo diferente  E) Otros   
 C) Un equipo dedicado a pruebas

Si marcó el literal E), por favor indique:

-----

6.7 Qué tipo de pruebas realizan?

*(Señale con una X todas las que aplican)*

- A) Pruebas de defectos  E) Caja negra\*   
 B) Pruebas de integración  F) Caja blanca\*\*   
 C) Pruebas orientas a objetos  G) Pruebas de stress   
 D) Pruebas unitarias  H) Otros

Caja negra\* Se prueba la funcionalidad del sistema.

Caja blanca\*\* Minucioso examen de detalles de los procedimientos.

Si marcó el literal H), por favor indique:

-----

6.8 Realizan pruebas de cobertura de código fuente\*?

(Marque con una X sólo una opción)

Siempre  Casi Siempre  Regularmente  Pocas veces  Nunca

\* **Cobertura de código fuente** Se refiere a la "*generación de histogramas*" que indiquen la frecuencia de ejecución de las líneas de código fuente usando datos de prueba, con el propósito de optimizarlo y determinar pruebas adicionales que ejecuten todo el código.

**6.9** Qué porcentaje de defectos encuentran en las pruebas?

(Del total de defectos en todo el desarrollo del software).

(Marque con una X sólo una opción)

[0-25%)  [25-50%)  [50-75%)  [75%-100%)  Desconoce

**6.10** Cuando en las pruebas se detectan muchos defectos en el proyecto, la empresa opta por:

(Señale con una X todas las que aplican)

- A) Modificar el tiempo planificado  C) No hacer modificaciones   
 B) Asignar más personal  D) Otro

Si marcó el literal D), por favor indique:

.....

## 7. MANTENIMIENTO

**7.1** Seleccione y priorice los factores que UD. considera son la principal causa del aumento de los costos de mantenimiento. (Para la principal causa 1, la siguiente 2 y así sucesivamente).

(Indique sólo los que aplican)

- A) Rotación del personal \_\_\_\_\_  
 B) Responsabilidad contractual \* \_\_\_\_\_  
 C) Habilidades del personal \_\_\_\_\_  
 D) Edad y Estructura del programa \*\* \_\_\_\_\_  
 E) Falta de documentación \_\_\_\_\_  
 F) Pruebas insuficientes antes de entregar el software \_\_\_\_\_

**Responsabilidad contractual \*** El contrato para mantener un sistema por lo general está separado del contrato de desarrollo del sistema. El contrato de mantenimiento se le puede dar a una compañía diferente en lugar de al desarrollador original del sistema. (Ian Sommerville)

**Edad y Estructura del programa \*\*** Conforme el programa aumenta su edad, su estructura tiende a degradarse por los cambios, por lo que es más difícil comprenderlo y cambiarlo. (Ian Sommerville)

**7.2** Qué porcentaje de los recursos totales de la empresa se usan para dar mantenimiento a los productos existentes y qué porcentaje para desarrollo de nuevos productos de software?

\_\_\_\_\_ % para mantenimiento

\_\_\_\_\_ % para nuevos productos

7.3 Indique el porcentaje de defectos sin corregir\* en el software entregado al cliente

Defectos \_\_\_\_\_%      Desconoce la cantidad

\* **Defectos sin corregir.**- Fallas que podrían presentarse en situaciones especiales o poco comunes en la operación del software pero que son conocidas por los desarrolladores.

7.4 Corrige y registra los errores\* reportados por el cliente?

*(Marque con una X solo una opción)*

Siempre     Casi Siempre     Regularmente     Pocas veces     Nunca

**Errores\*** Fallas que presenta el software cuando esta siendo usado por el cliente.

7.5 Registra el tiempo que se toma en corregir los errores reportados por el cliente?

*(Marque con una X solo una opción)*

Siempre     Casi Siempre     Regularmente     Pocas veces     Nunca

7.6 Se han perdido o dañado los datos del cliente por errores remanentes en el software

*(Marque con una X solo una opción)*

Siempre     Casi Siempre     Regularmente     Pocas veces     Nunca

7.7 Corrige y registra las solicitudes de cambios pedidas por el cliente?

*(Marque con una X solo una opción)*

Siempre     Casi Siempre     Regularmente     Pocas veces     Nunca

7.8 Registra el tiempo que se toma en hacer los cambios solicitados por el cliente?

*(Marque con una X solo una opción)*

Siempre     Casi Siempre     Regularmente     Pocas veces     Nunca

7.9 Indique el porcentaje de reinstalaciones parciales o completas del sistema que usualmente son requeridas para corregir errores de funcionamiento:

reinstalaciones parciales \* \_\_\_\_\_%      Desconoce los %s   
reinstalaciones completas \*\* \_\_\_\_\_%      Desconoce los %s

\* El software manifiesta errores que se solucionan al re-instalar el modulo(s) que presenta estos problemas.

\*\* El software manifiesta errores que se solucionan solamente al re-instalar completamente el software.

7.10 Seleccione y priorice los tipos de mantenimiento de software (luego de liberarlo) en base a la realidad de su empresa. (Para el más común 1, el siguiente 2 y así sucesivamente).

*(Indique solamente los que aplican)*

	Mantenimiento	Proyectos:		
		Pequeños	Medianos	Grandes
A)	Para reparar fallas del software (correctivo)	_____	_____	_____
B)	Para agregar o mejorar la funcionalidad del sistema (perfectivo)	_____	_____	_____
C)	Para adaptar el software a diferentes entornos operativos (adaptivo)	_____	_____	_____
D)	Para facilitar el mantenimiento futuro del sistema (preventivo)	_____	_____	_____

**GRACIAS POR SU COLABORACION**



# Apéndices

# PLAN DE MEDICIÓN

## 1. OBJETIVO

El objetivo de este documento es el de dar a conocer la forma como deben ser recopiladas, e interpretadas las métricas que se utilizarán para la **caracterización** y **seguimiento** del proceso de desarrollo de software principalmente desde el punto de vista del administrador de proyectos.

Ya que es reciente el uso de métricas en las actividades relacionadas al desarrollo del software, se busca que los resultados obtenidos proporcionen el **conocimiento** y la **comprensión** acerca de la forma como las prácticas actuales de la empresa proporcionan los productos y/o servicios que sus clientes requieren determinando cualitativa y cuantitativamente aspectos que se identificaron como relevantes y merecedores de atención durante todo el ciclo de desarrollo de los proyectos de software.

Este documento está formado por los siguientes apartados:

- Datos básicos para la recopilación de indicadores
- Objetivos de medición, preguntas, indicadores y datos a recopilar.
- Formularios de registro de datos
- Definición de datos (DF)
- Referencia para el ingreso de valores válidos (RF)

**DATOS BASICOS A RECOPIRAR PARA LA FORMACIÓN DE INDICADORES**

Cod	DATOS	Referencia a: Definición de terminos (DF) y guías para reportar (GR)	INDICADORES													DISPONIBILIDAD	FUENTE									
			Satisfacción del Cliente / Usuario V.S. Horas de participación en el proyecto	Error en la estimación del esfuerzo	Disponibilidad del personal	Esfuerzo empleado en la documentación	Complejidad	Avance del proyecto	Índice de rendimiento de costo	Índice de rendimiento de tiempo	Varianza del costo del proyecto	Inspecciones durante el proyecto	Variación en los requerimientos	Tiempo de aprendizaje de componentes	Tamaño de los componentes de SW			Defectos v.s. Fallas	Diagrama Causa-efecto							
1	Proyecto	1	X	X														Disponibile	Forma F00, Orden de trabajo							
2	Requerimientos	2, 3, 4																X	Obtenida con poco esfuerzo	Forma F00, Lista de requerimientos						
3	Aplicación de software	5																	Disponibile	Forma F00						
4	Proyecto de software	1						X											Disponibile	Forma F00						
5	Satisfacción cliente	6	X																Obtenida con poco esfuerzo	Forma F00, Encuesta cliente						
6	Satisfacción usuario	7, 8	X																Obtenida con poco esfuerzo	Forma F00, Encuesta usuario						
7	Fecha inicio	9																	X	Disponibile	Forma F00, Project					
8	Fecha fin	10	X																X	Disponibile	Forma F00, Project					
9	Personal estimado	11, 12, 13																		X	Disponibile	Forma F00, Project				
10	Personal real	12, 13																		X	Disponibile	Forma F00, Project				
11	Costo estimado	11, 14																			X	Disponibile	Forma F00, Excel			
12	Costo real	14																			X	Disponibile	Forma F00, Excel			
13	Complejidad negocio	15																			X	Obtenida con poco esfuerzo	Forma F00, Encuesta de Comp. Tec.			
14	Complejidad técnica	15																			X	Obtenida con poco esfuerzo	Forma F00, Encuesta de Comp. Neg.			
15	Rol	16																					Disponibile	Forma F00		
16	Nombre	GR																						Disponibile	Forma F00	
17	Experiencia desarrollo	17																						Disponibile	Forma F00, Curriculum vitae	
18	Experiencia negocio	17																						Disponibile	Forma F00, Curriculum vitae	
19	Rotacion	18																							Obtenida con poco esfuerzo	Forma F00
20	Tiempo	19																							Obtenida con poco esfuerzo	Forma F00
21	Causa	19																							Obtenida con poco esfuerzo	Forma F00

Figura No. 1.



**DATOS BASICOS A RECOPIRAR PARA LA FORMACIÓN DE INDICADORES**

Cod	DATOS	Referencia a: Definición de terminos (DF) y guías para reportar (GR)	INDICADORES												DISPONIBILIDAD	FUENTE				
			Satisfacción del Cliente / Usuario V.S. Horas de participación en el proyecto	Error en la estimación del esfuerzo	Disponibilidad del personal	Esfuerzo empleado en la documentación	Complejidad	Avance del proyecto	Índice de rendimiento de costo	Varianza del costo del proyecto	Inyecciones durante el proyecto	Complejidad en los requerimientos	Tiempo de aprendizaje de componentes	Tamaño de los componentes de SW						
40	Número de reporte	GR											X					Disponible	Forma F02	
41	Fecha	GR											X					Disponible	Forma F02	
42	Estado	GR															X	Disponible	Forma F02	
43	Fecha abierto	GR																Disponible	Forma F02	
44	Tipo	31, 32															X	Disponible	Forma F02	
45	Severidad	GR															X	Disponible	Forma F02	
46	Esfuerzo estimado	11, 30															X	Disponible	Forma F02	
47	Esfuerzo real	30															X	Disponible	Forma F02	
48	Requerimiento	3, 4, GR											X					Obtenida con poco esfuerzo	Forma F02	
49	Funcional / No funcional	3, 4											X					Disponible	Forma F02	
50	Metodo que detecto el problema	33, GR															X	Disponible	Forma F02	
51	Observaciones	GR															X	Disponible	Forma F02	
52	Descripción del problema	GR																X	Disponible	Forma F02
53	Componente afectado	GR																X	Disponible	Forma F02
54	Causas	GR																X	Disponible	Forma F02
55	Efectos	GR																X	Disponible	Forma F02
56	Fecha Mes	GR																	Disponible	Forma F03
57	Líneas de código	34															X		Difícil de obtener	Forma F03
58	Componente, método o precedimiento	35											X	X	X				Difícil de obtener	Forma F04
59	Horas de desarrollo del componente	GR																	Disponible	Forma F04
60	Número de defectos	31, 32, GR																	Obtenida con poco esfuerzo	Forma F04
61	Complejidad ciclomática	36											X	X			X		Difícil de obtener	Forma F04
62	Demora de aprendizaje	GR															X		Difícil de obtener	Forma F04

Figura No. 3



**Objetivo de medición No. 1:** Determinar si el nivel de satisfacción del cliente esta relacionado con su participación en el proyecto para limitar o promover su colaboración con el personal asignado al proyecto.

- Pregunta No. 1.** ¿Cuál es el nivel de satisfacción del cliente y/o usuario con el producto de software desarrollado?
- Pregunta No. 2.** ¿En que actividades durante el desarrollo del proyecto es necesaria la participación del cliente y/o usuario final?
- Pregunta No. 3.** ¿Cuál es la percepción del cliente con la metodología de desarrollo de proyectos de la empresa?

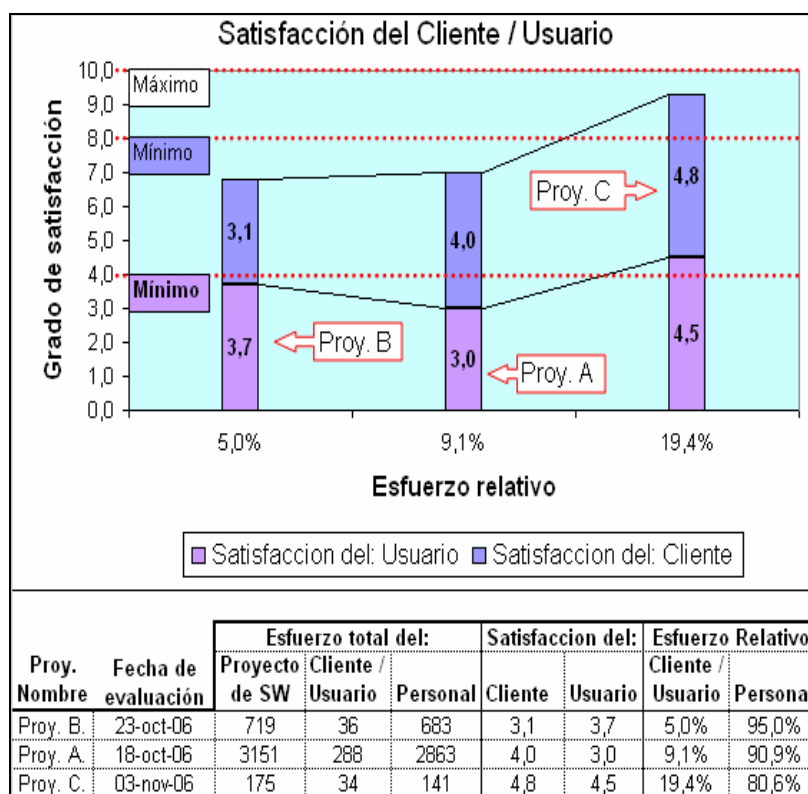


Figura No. 4. Participación del cliente / usuario VS. Grado de Satisfacción

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos derivados:</b>
Proy. Nombre.	1	F00	<b>A)</b> Esfuerzo total del proyecto de software.
Satisfacción del cliente.	5	F00	<b>B)</b> Esfuerzo total de los clientes y usuarios.
Satisfacción del usuario.	6	F00	<b>C)</b> Esfuerzo total del personal.
Esfuerzo real.	47	F01	<b>D)</b> Esfuerzo relativo de los clientes y usuarios.
Fecha de evaluación.	---	Encuesta	<b>E)</b> Esfuerzo relativo del personal.

**Cálculo(s):**

- A)** Esfuerzo total del proyecto de software = La suma los valores registrados en el campo No. 47 del Formulario F01 en todas las tareas realizadas y registradas.
- B)** Esfuerzo total del cliente y usuario = Suma del esfuerzo realizado (47, F01) solo por los Clientes y usuarios.
- C)** Esfuerzo total del personal = Suma del esfuerzo realizado (47, F01) por el personal de la empresa.
- D)** Esfuerzo relativo del, Cliente/Usuario = (B / A) \* 100.
- E)** Esfuerzo relativo del, personal = (C / A) \* 100.

### **Análisis del indicador:**

Este indicador busca determinar si existe alguna correlación entre la participación de los clientes y usuarios en el ciclo de desarrollo del software con la satisfacción que consiguen respecto del producto que reciben y con la forma actual de trabajar de la empresa. Se presume una correlación positiva entre estos dos factores, ya que a mayor participación de clientes y usuarios en el desarrollo de los proyectos, mayor es el nivel de satisfacción alcanzado.

La figura No. 4 es el tipo “columna apilada” y ella se observa los valores obtenidos de satisfacción de los clientes y usuarios así como el esfuerzo que estos emplearon durante el desarrollo de sus respectivos proyectos de software.

En el eje horizontal, se observa el esfuerzo de participación relativo de clientes y usuarios, mientras que en el eje vertical se desglosa el valor de satisfacción evaluado a través de las respectivas encuestas diseñadas con este propósito y aplicadas al finalizar el proyecto de software.

Las líneas punteadas de color rojo, representan puntos de referencia con los que se juzga si se ha alcanzado un nivel de satisfacción previamente determinado. En este caso, escogemos el valor 4 como el valor mínimo a alcanzarse tanto en la satisfacción del cliente como en la de los usuarios.

De la figura de ejemplo se interpreta que el proyecto C ha sobrepasado los valores de referencia mínimos del eje vertical por lo que podemos decir que los clientes y usuarios de este proyecto se encuentran en el intervalo de satisfacción (entre 8 y 10 del eje vertical) deseado por la empresa. En el caso de los proyectos A y B, a excepción del cliente del proyecto A, estos no llegan al valor mínimo de satisfacción que la empresa desea alcanzar.

**Objetivo de medición No. 2:** Establecer el esfuerzo adecuado que se requerirá del personal asignado al proyecto para mantener un nivel sostenido de rendimiento entre sus miembros.

<b>Pregunta No. 1</b> ¿Se dispone del personal adecuado para el desarrollo del proyecto?
<b>Pregunta No. 2</b> ¿El esfuerzo requerido al personal asignado al proyecto es adecuado con la carga de trabajo que tienen actualmente?

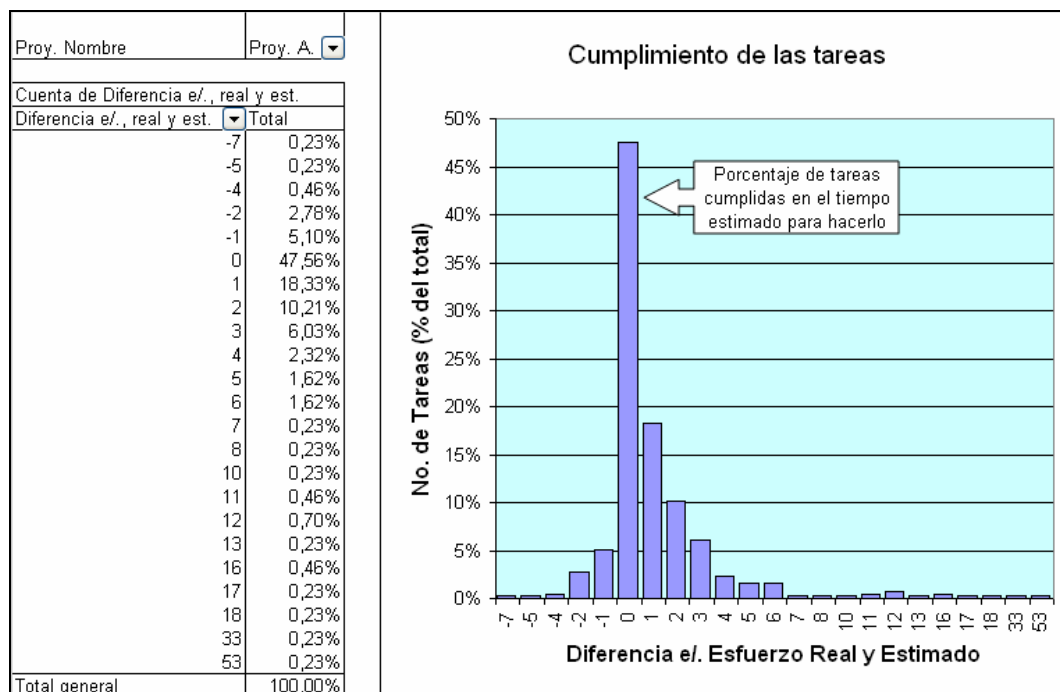


Figura No. 5. Tabla dinámica error en la estimación del esfuerzo.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos derivados:</b>
Proy. Nombre.	1	F00	<b>F)</b> Diferencia e/., esfuerzo real y estimado
Esfuerzo estimado por tarea	46	F01	<b>G)</b> Error [%]
Esfuerzo real por tarea.	47	F01	

Proy. Nombre	Proy. A. ▾														
Suma de Esfuerzo real	Nombre, ▾														
Rol	PA01	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	UA01	UA02	UA03	UA04	Total general
Cliente - Developer Oracle												10			10
Cliente - Gerente de Sistemas														22	22
Cliente - Usuario															59
Gerente de Proyectos									39						39
Cliente - Developer FoxPro											78				78
Cliente - Jefe de Sistemas														119	119
Jefe de proyecto										518					518
Developer .net	417	460													877
Developer Oracle			528	547	164	47	142	1							1429
Total general	417	460	528	547	164	47	142	1	39	518	78	10	22	119	3151

Figura No. 6. Tabla dinámica esfuerzo del personal por rol.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos derivados:</b>
Proy. Nombre.	1	F00	<b>H)</b> Esfuerzo real del personal por rol.
Rol.	15	F00	
Nombre, Apellido.	16	F00	
Esfuerzo real por persona	39	F01	

**Cálculo(s):**

<b>F)</b> Diferencia e/. esfuerzo real y estimado = Esfuerzo real empleado en cada tarea (46, F01) – Esfuerzo estimado (47, F01).
<b>G)</b> Error [%] = (Diferencia e/. esfuerzo real y estimado) / Esfuerzo estimado.
<b>H)</b> Esfuerzo real del personal por rol = Suma del esfuerzo real (39, F01) de cada miembro del equipo de desarrollo según el rol que desempeña en el proyecto usando la tabla dinámica de MS. Excel.

**Análisis del indicador:**

Error en la estimación del esfuerzo (**Figura No. 5**).

A través de esta tabla dinámica y su correspondiente gráfico se pueden apreciar la cantidad de tareas (en porcentaje respecto del total) que se desarrollan a tiempo, con retraso, o de forma anticipada.

La columna “total”, contabiliza automáticamente la cantidad de tareas que se cumplieron a tiempo y que corresponden al valor cero de la columna “Diferencia e/., real y est”. En esta misma columna, los valores negativos indican las horas-hombre que un porcentaje de tareas ahorraron mientras que los valores positivos indican la cantidad de horas-hombre adicionales se utilizaron para terminar sus respectivas tareas.

En la figura de ejemplo, se puede apreciar que de todas las tareas realizadas hasta la fecha 47.56% se han terminado a tiempo, 8.82% antes de lo estimado y 18.33% después de lo estimado.

Finalmente, el error en la estimación del esfuerzo necesario para realizar las diferentes tareas del proyecto, se calcula solamente para las tareas que requieren más esfuerzo en ser completadas.

Esfuerzo del personal por rol (**Figura No. 6**).

En esta tabla se observa como se ha distribuido el esfuerzo total empleado por el personal, clientes y/o usuarios según el rol(es) desempeñado(s), permitiendo identificar fácilmente los recursos humanos de mayor importancia para el desarrollo del proyecto.

La celda inferior derecha muestra el esfuerzo total consumido en el proyecto (Ej. 3151 horas-hombre) y el mismo que se encuentra desglosado en la columna “total general” que contabiliza el esfuerzo empleado por rol, y en la fila “total general” que contabiliza el esfuerzo que cada persona ha aportado al proyecto. Las etiquetas “PA01”, “PA03”, “UA01”, etc., simplemente son códigos que reemplazan el nombre y apellido de los miembros del personal y usuarios siendo la decisión de proceder de esta forma, de exclusiva responsabilidad del administrador de proyectos.

**Objetivo de medición:** Determinar el esfuerzo empleado en realizar la documentación durante el desarrollo del proyecto para estimar el esfuerzo que sería necesario utilizar en una adecuada documentación en proyectos similares

<b>Pregunta No. 1.</b> ¿Qué documentos se generaron durante el desarrollo del proyecto?
<b>Pregunta No. 2.</b> ¿Cual es el esfuerzo utilizado en la documentación del proyecto?

Proy. Nombre	Proy. A.	
Suma de Esf. Documentacion		
Paginas	Documento	Total
1	Acta de Reunión Informe de Actividades Diaras	1 1
2	Acta de Reunión Actas de Entrega de Módulo de Sobrestadía. Cambios y Definiciones establecidas en la Charla Copia de Apuntes y Reporte de Actividades diarias Reporte de errores encontrados. Solicitudes varias...	2 2 2 2 1 1
3	Acta de Reunión Cronograma de Cobranzas	6 1
Total general		78

Figura. 7. Tabla dinámica esfuerzo empleado en documentación.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos derivados:</b>
Proy. Nombre.	1	F00	I) Porcentaje de esfuerzo empleado en documentación.
Documentación	27	F01	
Páginas	26	F01	
Esfuerzo real por tarea	47	F01	

**Cálculo(s):**

I) Porcentaje de esfuerzo empleado en documentación = $\frac{\text{Suma del esfuerzo real (47, F01)}}{\text{Esfuerzo total del proyecto de software}}$
--

**Análisis del indicador:**

Esfuerzo empleado en documentación (**Figura No. 7**)

En esta tabla se presentan los diferentes documentos que se han elaborado durante el desarrollo del proyecto, junto con el respectivo número de páginas y horas-hombre utilizadas que dan una idea del tamaño del documento y esfuerzo empleado en su realización.

En la celda inferior derecha se contabiliza el esfuerzo total utilizado en la elaboración de los documentos, aunque por conveniencia (de espacio) se han presentado solamente aquellos documentos de hasta 3 páginas de extensión.

La determinación de un porcentaje adecuado de esfuerzo que se debería considerar para documentar proyectos similares entre si, se logrará al comparar la disponibilidad y conveniencia de dicha documentación en el mantenimiento o posteriores modificaciones a la aplicación de software.

**Objetivo de medición:** Monitorear el avance de las actividades y tareas planificadas para cumplir o reconsiderar los compromisos establecidos al inicio del proyecto

<b>Pregunta No. 1.</b> ¿Actividades y tareas se planificaron teniendo en cuenta la complejidad del proyecto?
<b>Pregunta No. 2.</b> ¿Las actividades y tareas planificadas para el proyecto se cumplen según el cronograma establecido?
<b>Pregunta No. 3.</b> ¿Las actividades y tareas se completan sin excederse del presupuesto establecido al inicio del proyecto?

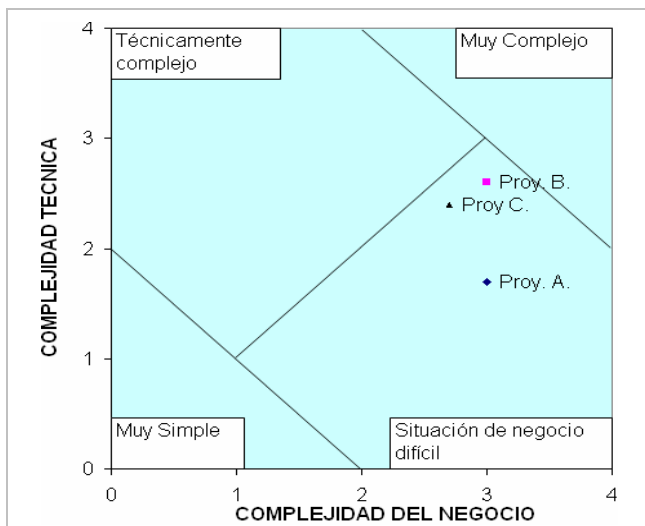


Figura 8. Complejidad estimada de proyectos

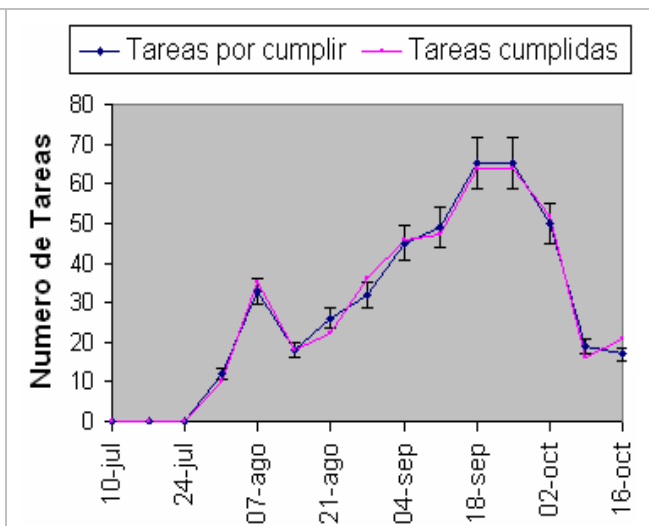


Figura 9. Avance del proyecto.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>		<b>Datos derivados:</b>
Proy. Nombre.	1	F00	Fig. 8	<b>J)</b> Tareas terminadas. (Fig. 9.)
Complejidad de negocio.	13	F00	Fig. 8	<b>K)</b> Tareas por cumplir. (Fig. 9)
Complejidad técnica.	14	F00	Fig. 8	
Fecha de fin real	33	F01	Fig. 9	

Proy. Nombre	Proy. A.								
Suma de PV									
Fecha fin real	Fecha fin est.	PV	EV	AC	CV	SV	CPI	SPI	
Fecha eval	28/07/2006	165,16	78,5	68,9	9,6	-86,66	1,14	0,48	
	06/08/2006	790,97	739,77	796,18	-56,41	-51,20	0,93	0,94	
	19/08/2006	1775,12	1694,32	1734,72	-40,4	-80,80	0,98	0,95	
	03/09/2006	3051,47	3069,22	3418,98	-349,76	17,75	0,90	1,01	
	15/09/2006	4433,91	4422,53	5018,41	-595,88	-11,38	0,88	1,00	
	29/09/2006	6096,96	5901,44	6762,22	-860,78	-195,52	0,87	0,97	
	13/10/2006	6702,75	6702,75	7760,43	-1057,68	0,00	0,86	1,00	

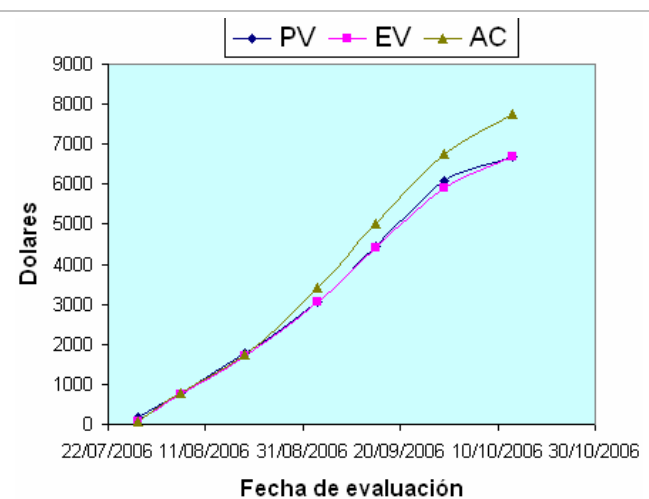


Figura 10. Tabla dinámica (EVMS) sistema de administración por valor devengado.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>
Proy. Nombre.	1	F00	Fecha real de fin	33	F01
Fecha estimada fin	31	F01	Costo estimado	36	F01
			Costo real	37	F01

**Datos derivados:**

- L) El costo presupuestado para el trabajo planificado. (Planned Value)
- M) El costo presupuestado para el trabajo realizado. (Earned Value)
- N) El costo real del trabajo realizado. (Actual Cost)
- O) Varianza del costo. (Cost Variance)
- P) Varianza del calendario. (Schedule Variance)
- Q) Índice de rendimiento de costo (Cost Performance Index)
- R) Índice de rendimiento de tiempo. (Schedule Performance Index)

**Cálculo(s):**

**J)** Tareas terminadas = Suma de todas las tareas terminadas hasta la fecha de evaluación del proyecto. (la fecha de evaluación de un proyectos es determinada a juicio y conveniencia del administrador de proyectos)

**K)** Tareas por cumplir. = Suma de todas las tareas que debían haberse terminado hasta la fecha de evaluación del proyecto.

**L)** Planned Value (PV) = La suma de los presupuestos autorizados para todo el trabajo planificado que debe ser cumplido dentro de un periodo específico de tiempo. (Desde el inicio hasta la fecha de evaluación del proyecto) [Gopal, pág. 397]

**M)** Earned Value (EV) = La suma de los presupuestos autorizados para el trabajo realmente desarrollado dentro de un periodo específico de tiempo. [Gopal, pág. 397]

**N)** Actual Cost (AC) = El costo real incurrido, hasta la fecha de evaluación del proyecto.

**O)** Cost Variance (CV) = La diferencia entre el costo presupuestado de una actividad y el costo real (AC) de esa actividad;  $CV = EV - AC$ . [Gopal, pág. 397]

**P)** Schedule Variance (SV) = La diferencia numérica entre el valor devengado (EV) y el valor planificado (PV).

**Q)** Cost Performance Index (CPI) = La tasa de eficiencia del costo entre el valor devengado (EV) y el costo real (AC);  $CPI = EV / AC$ . [Gopal, pág. 397]

**R)** Schedule Performance Index (SPI) = El índice de eficiencia del cronograma del cumplimiento del valor devengado (EV) contra el valor planificado;  $SPI = EV/PV$ . [Gopal, pág. 397]

**Análisis del indicador:**

Complejidad estimada de los proyectos (**Figura 8**)

Este indicador busca categorizar los proyectos evaluados en cuatro categorías: 1) Muy simple, 2) técnicamente complejo, 3) situación de negocio difícil, y 4) muy complejo dependiendo de los valores estimados de complejidad de negocio y complejidad técnica de los proyectos.

1. Muy simple: Los proyectos de esta categoría son de baja complejidad – proyectos rutinarios o simples.
2. Técnicamente complejo: Esta categoría pertenece a proyectos con alta complejidad técnica.
3. Situación de negocio difícil: Esta categoría corresponde a proyectos con complejidad de negocio alta.
4. Muy complejo: Los proyectos de esta categoría son altamente complejos – de gran complejidad técnica y de negocio. [Gopal, pág. 138]

“El comprender la complejidad de los proyectos nos ayuda a seleccionar a los adecuados patrocinadores, un administrador de proyectos y el equipo de trabajo, así como obtener una primera apreciación de los riesgos inherentes al proyecto”. [Gopal, pág. 142] En la figura de ejemplo, se observan tres proyectos cuya

complejidad esta clasificada como “situación de negocio difícil”, con valores de complejidad de negocio altos y similares pero con niveles de complejidad técnica diferente.

#### Avance del proyecto (**Figura 9**)

Este indicador provee información acerca de “lo bien” que se está desarrollando el proyecto de software tomando en cuenta el cumplimiento de las tareas que se planificaron cumplir dentro de un periodo de tiempo específico.

Para contar con los datos necesarios para formar este indicador se requiere una planificación inicial del proyecto asignado una fecha de entrega a cada una de la tareas y/o actividades de este proyecto, lo que permite trazar una línea de referencia para el cumplimiento de las tareas (línea azul) De forma periódica se hace el seguimiento a cada tarea del proyecto registrando la fecha real de finalización lo que permite establecer una línea de cumplimiento real y hace posible una comparación entre el avance estimado y el avance real del proyecto.

En cada punto de evaluación del estado del proyecto, se hace una comparación entre el número de tareas planificadas a cumplir y el número de tareas realmente cumplidas. Se permite un margen de error, el cual es escogido a criterio del administrador y que esta representado por las líneas verticales sobre la línea azul de las tareas por cumplir de la figura 9.

**Nota:** La figura de ejemplo corresponde a un proyecto que ha sufrido retrasos en su desarrollo que sin embargo no se han reflejado en este gráfico. Esto es debido a que no existía una planificación inicial del proyecto. Por consiguiente se debe tener cuidado en la interpretación de este indicador cuando no se tiene la planificación de las tareas por cumplir a largo plazo.

#### (EVMS) Sistema de Administración por Valor Devengado (**Figura 10**)

El “Sistema de Administración por Valor Devengado”, (EVMS) esta diseñado para seguir el progreso del proyecto. EVMS es usado para:

- Determinar el valor (dólares) del trabajo completado hasta la fecha de evaluación del proyecto.
- Medir el progreso al comparar el valor en dólares del trabajo completado con los dólares que ha planificado gastar y los dólares que realmente ha gastado. [Gopal, pág. 396]

Los datos observados en la figura 10, indican que el proyecto esta avanzando según lo planificado ( $EV \approx PV$ ), sin embargo el costo real se incrementa constantemente y esta por encima de lo presupuestado ( $AC > PV$ ).

**Cost Performance Index (CPI).** El CPI provee una medida de la tasa a la que se consume el dinero en el proyecto, medida hasta la fecha de evaluación del estado del proyecto. En términos matemáticos:  $CPI = EV / AC$ . Un CPI igual a uno indica que el costo del proyecto es igual al planificado; menor a uno indica que el costo del proyecto es mayor al planificado; y mayor a uno indica que el costo es menor al planificado. [Gopal, pág. 398]

**Schedule Performance Index (SPI).** El SPI provee una medida de la tasa a la cual el trabajo es cumplido, medido hasta la fecha de evaluación del estado del proyecto. En términos matemáticos:  $SPI = EV/PV$ . Un SPI igual a uno indica que el proyecto se desarrolla de acuerdo al cronograma; menor a uno indica que esta atrasado en cronograma; y mayor a uno indica que adelanto en el cronograma. [Gopal, pág. 398]

[Gopal] Gopal K. Kapur. Project Management for Information, Technology, Business, and Certification. 2005 Pearson Prentice Hall.



**Objetivo de medición:** Estimar la cantidad adecuada de inspecciones sobre los diseños (actividades, tareas y/o entregables) para disminuir la propagación y severidad de los defectos.

<b>Pregunta No. 1.</b> ¿Cuántas inspecciones (revisiones) se efectúan por actividad (no solo de diseño) y/o entregables del proyecto?
<b>Pregunta No. 2.</b> ¿Qué diferencia se aprecia entre proyectos que tienen inspecciones regulares y aquellos proyectos con pocas inspecciones?

Proy. Nombre	Proy. A.		
Suma de Esfuerzo real.			
Fase	Nombre tarea	Tarea Id.	Total
Construcción	1aRevisión de Operaciones de Consolidadora	A0028	22
	2aRevisión de Módulos de Operaciones Consolidadora	A0030	6
	3aRevisión de Módulos de Operaciones Consolidadora	A0032	5
	Reunión de revisión de cronogramas	A0125	2
	Reunión para revisión de Avances del Proyecto	A0232	7
	Reuniones para revisión de últimos cambios	A0098	8
	Revisando cambios en Factura de Cuentas	A0267	8
	Revisando la generación de Centros de Costo	A0213	17
	Revisión de funcionalidades implementadas	A0271	10
	Revisión de Funcionalidades y Configuraciones	A0069	1
	Revisión de la funcionalidad implementada	A0263	9
	Revisión de los Módulos de Compras e Inventarios	A0077	2
	Total general		

Figura 11. Tabla dinámica inspecciones por actividad o entregable generado.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos derivados:</b>
Proy. Nombre.	1	F00	<b>S)</b> Suma de esfuerzo en inspecciones.
Tarea Id.	23	F01	<b>T)</b> Porcentaje del total.
Fase	22	F01	
Esfuerzo real	47	F01	

**Cálculo(s):**

<b>S)</b> Suma de esfuerzo en inspecciones = Suma del esfuerzo (47, F01) de las tareas identificadas como inspecciones y/o revisiones.
<b>T)</b> Porcentaje del total = Suma de esfuerzo en inspecciones / Esfuerzo total del proyecto de software.

**Análisis del indicador:**

Inspecciones por actividad o entregable generado (**Figura. 11**)

Este indicador resume el esfuerzo empleado en actividades de control realizadas sobre los entregables y/o tareas realizadas al proyecto de software en sus diferentes fases de desarrollo.

La celda inferior derecha de la tabla indica el esfuerzo total empleado en inspecciones, aunque por conveniencia (de espacio) se han presentado solamente las actividades de control realizadas en la fase de construcción.

La determinación de un porcentaje adecuado de esfuerzo que se debería considerar para realizar estas actividades de control que permitan obtener resultados satisfactorios entre proyectos similares será posible al relacionar el nivel alcanzado de satisfacción de clientes/usuarios con la cantidad de esfuerzo empleado en inspecciones (similar al indicador Participación del cliente / usuario VS. Grado de Satisfacción).

**Objetivo de medición** Monitorear los cambios que se dan en los requerimientos funcionales del software durante el desarrollo del proyecto para determinar su efecto sobre el cronograma y presupuesto del proyecto

<b>Pregunta No. 1.</b> ¿El cliente define concisa y completamente todos sus requerimientos en las fases iniciales o de especificación del proyecto?
<b>Pregunta No. 2.</b> ¿Los requerimientos del proyecto se mantienen estables (sin cambios) durante el desarrollo del mismo?

Perfil de proyectos		Empresa 1		Empresa 2		Empresa 3	
<b>Nombre del proyecto:</b>		Proy. A.		Proy. B.		Proy. C.	
<b>Tipo de:</b>	Aplicación	Administrativa, Financiera.		Workflow		Portal Web	
	Proyecto SW	Nuevo		Nuevo		Nuevo	
<b>Fecha de:</b>	Inicio	01-may-05		28-nov-05		06-jun-06	
	Fin	12-dic-06, "Activo"		15-dic-06, "Activo"		20-oct-06, "Acabado"	
<b>Número de personas asignadas:</b>	Estimado	5		2		1	
	Real	9		5		1	
<b>Costo [\$]:</b>	Estimado	Reservado		Reservado		Reservado	
	Real	Reservado		Reservado		Reservado	
<b>Esfuerzo [horas x hombre ]:</b>	Cliente / Usuario	9,1%	3151	5%	719	19,4%	175
	Personal	90,9%		95%		80,6%	
<b>Satisfacción [escala Likert]:</b>	Cliente	4		3,1		4,8	
	Usuario	3		3,7		4,5	
<b>Complejidad [escala Likert]:</b>	Negocio	3		3		2,7	
	Técnica	1,66		2,6		2,4	
<b>Nro. de Requerimientos:</b>	Iniciales	14		23		14	
	Agregados	6	45	5	22	0	2
	Modificados	39		17		2	
	Eliminados	0		0		0	
Semanas interrumpido	1	5		6			
<b>Interrupciones:</b>	Causa y tiempo interrumpido [semanas]	Por capacitación al nuevo personal asignado al proyecto	1	No se contaba con personal necesario para el desarrollo	5	El personal realizó tareas de soporte. Por pedido del cliente.	2
							4

Figura 12. Perfil de proyectos.

Datos registrados:	Cod,	Fuente	Datos derivados:
Proy. Nombre.	1	F00	U) Variación en los requerimientos.
Requerimientos iniciales	2	F00	V) Requerimientos agregados.
Requerimiento	48	F02	W) Requerimientos modificados.

**Cálculo(s):**

U) Variación en los requerimientos = Nro. de requerimientos agregados + Nro. de requerimientos modificados. (Figura. 12)
V) Requerimientos agregados = Nro. de requerimientos agregados / Nro. requerimientos iniciales.
W) Requerimientos modificados = Nro de requerimientos modificados / Nro. requerimientos iniciales.

**Análisis del indicador:**

En el formulario F02 se especifica que tipo de cambio se reporta y con esta información se contabiliza y se presentan estos valores como parte de la figura 12 en la que se presenta un perfil del proyecto con información consolidada de métricas como esfuerzo, costo, tamaño del personal, etc., y que sirven para dar contexto y proporcionar una mejor visión del estado del proyecto. El propósito es mantener informado al administrador de proyectos de estos cambios de tal forma que pueda tomar decisiones oportunas en caso de observar más cambios de los que se pueden considerar “normales”.

**Objetivo de medición:** Conocer la ocurrencia y causa de los defectos durante el desarrollo de los proyectos de software para identificar acciones de prevención y/o contingencia.

<b>Pregunta No. 1.</b> ¿De qué forma se distribuyen los defectos durante el ciclo de desarrollo del proyecto?
<b>Pregunta No. 2.</b> ¿Que clase de defectos son los más recurrentes?
<b>Pregunta No. 3.</b> ¿Qué fracción del tiempo del proyecto se utiliza para corregir defectos?
<b>Pregunta No. 4.</b> ¿Cuán eficiente es la empresa en corregir defectos en el software antes de liberarlo?

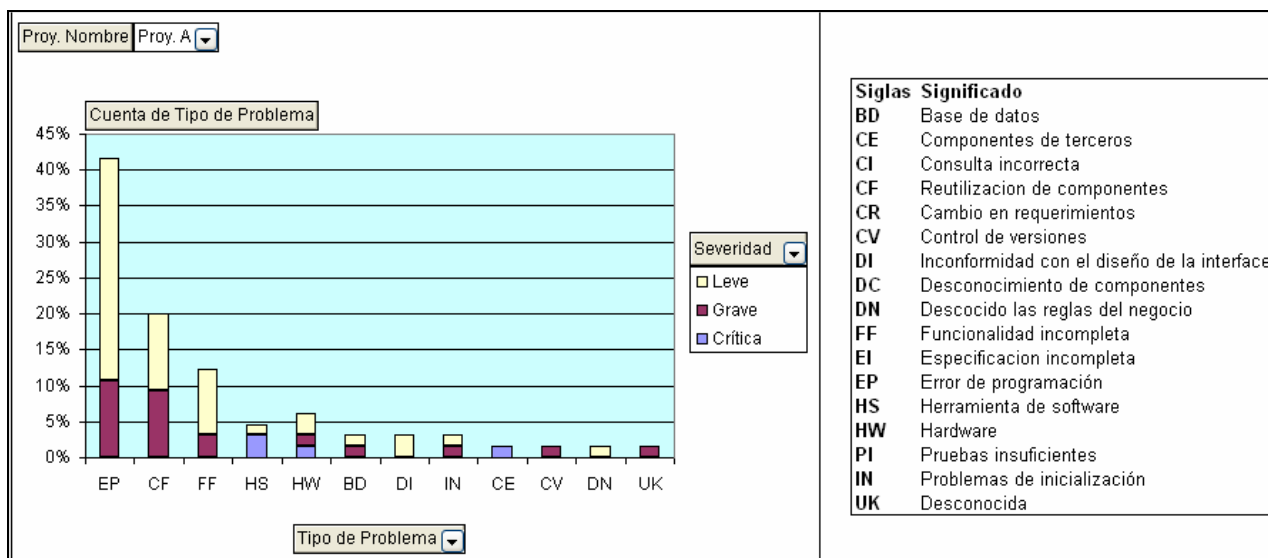


Figura 13. Pareto de defectos.

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos derivados:</b>
Proy. Nombre.	1	F00	X) Tipo de problema
Severidad	45	F02	Y) Cuenta de tipo de problema
Descripción del problema	52	F02	

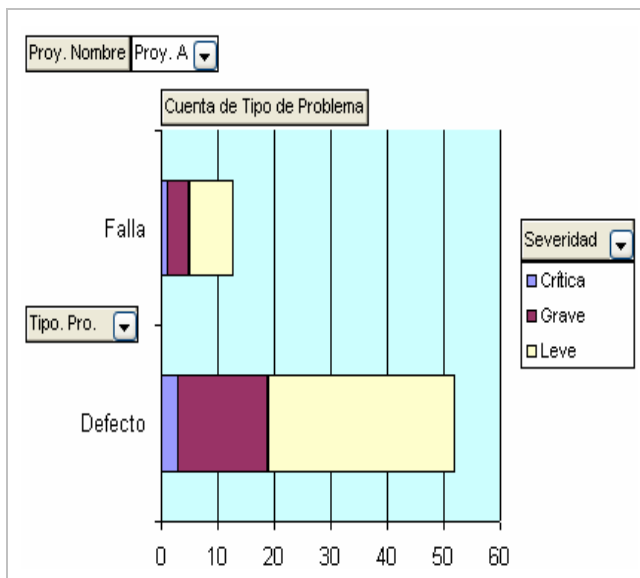


Figura 14. Defectos VS. Fallas.

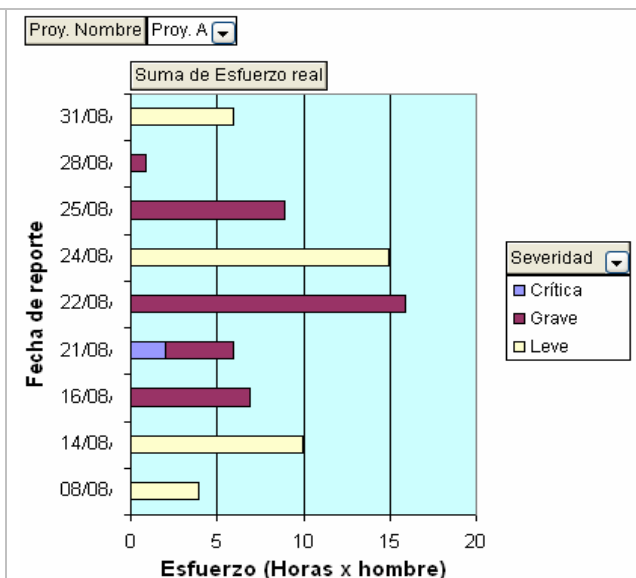


Figura 15. Esfuerzo empleado en correcciones

<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>	<b>Datos registrados:</b>	<b>Cod,</b>	<b>Fuente</b>
Proy. Nombre.	1	F00	Severidad	45	F02
Tipo de problema	X	Fig. 13	Fecha de reporte	41	F02
Esfuerzo real en corregir	47	F02			

**Datos derivados:**

<b>Z)</b> Esfuerzo en correcciones
------------------------------------

**Cálculo(s):**

<b>X)</b> Tipo de problema = Usando la descripción de los problemas reportados (52, F02) se procederá a clasificarlos conforme el criterio del administrador de proyectos para lo cual debe existir una lista de problemas conocidos como la mostrada en la figura 13.
<b>Y)</b> Cuenta tipo de problema = Con la ayuda de la tabla dinámica se contabilizan los problemas reportados por cada tipo de problema previamente determinado.
<b>Z)</b> Esfuerzo en correcciones = Suma del esfuerzo real (47, F01) empleado en solucionar problemas, clasificados según su fecha de reporte.

**Análisis del indicador:**

**Pareto de defectos. (Figura 13)**

Con la ayuda de este indicador, se busca conocer los problemas más frecuentes ocurridos durante el desarrollo del proyecto. En el análisis de Pareto, “se dice que, en muchos casos, la mayoría de los defectos y de su costo se deben a un número relativamente pequeño de causas” [Hitoshi, pág. 20].

Después de que cada problema que ha sido reportado con el formulario (F02) ha sido revisado y clasificado por el administrador de proyectos, la herramienta “tabla dinámica” de Microsoft Excel realizará el conteo de estos problemas agrupándolos por tipo de problema en el eje horizontal y clasificándolos por severidad en el eje vertical tal y como se observa en la figura 13.

En la figura de ejemplo se observa que los problemas más frecuentes (que consideraremos como los “pocos vitales”) son los relacionados a los errores en la programación (EP), la reutilización de componentes y funcionalidades incompletas en la aplicación de software y que totalizan el 74% de todos los problemas que han sido reportados.

[**Hitoshi**] Hitoshi Kume. Herramientas estadísticas básicas para el mejoramiento de la calidad. 2002. Grupo editorial Norma.

**Defectos VS. Fallas (Figura 14)**

Para el análisis de este indicador se definen los términos “defecto” y “falla” de la siguiente manera:

**Defecto:** Algún tipo de problema que se presentó durante el desarrollo del proyecto causando: 1) mal funcionamiento de la aplicación de software, 2) re-trabajo y/o 3) demoras en la entrega del producto.

**Falla:** Hace mención a interrupciones inesperadas en el funcionamiento de la aplicación de software cuando está siendo utilizada por el cliente.

Para calcular el porcentaje que las fallas representan respecto del total de todos los problemas reportados durante el desarrollo del proyecto y los primeros tres meses de uso de la aplicación se usan los valores contabilizados para defectos y fallas en la siguiente formula:

$$\text{Fuga de defectos} = \text{Fallas} / (\text{Defectos} + \text{Fallas});$$

En el gráfico de la figura 14, se grafican 52 defectos y 13 fallas (Fuga de defectos = 20%), y en el que se resalta que son pocos los defectos y fallas clasificados con una severidad critica para satisfacción de clientes y 7 usuarios. Las actividades de control de calidad deberán estar dirigidas a detectar y corregir la mayor cantidad de defectos en el software antes de su entrega formal.





**FORMULARIO F02**

REPORTE DE PROBLEMAS O CAMBIOS EN REQUERIMIENTOS (F02)									
<b>PROYECTO</b>					<b>Número de REPORTE</b>				
1					40				
<b>FASE</b>					<b>FECHA</b>				
22					41				
<b>ESTADO</b>		<b>TIPO</b>		<b>SEVERIDAD</b>		<b>ESFUERZO</b>		<b>REQUERIMIENTO</b>	
En cola	42	Defecto	44	Leve	45	Estimado	46	Agregado	48
Abierto		Falla		Grave		Real	47	Modificado	
Fecha abierto	43	Otro		Crítica				Eliminado	
Cerrado								Funcional	49
Descartado								No Funcional	
<b>METODO QUE DETECTO EL PROBLEMA</b>									
Inspección	50	Caja Negra		Otro		Observaciones:			
Peer review		Caja Blanca		Otro		51			
P. Stress		Simulación		Otro					
<b>DESCRIPCION DEL PROBLEMA</b>					<b>COMPONENTE(s) AFECTADO</b>				
52					53				
<b>POR QUÉ SE PRODUJO? (CAUSAS)</b>					<b>IMPACTO EN EL PROYECTO? (EFECTOS)</b>				
54					55				





## DEFINICIÓN DE DATOS (DF)

### 1. Proyecto

Un conjunto administrado de recursos interrelacionados que entrega uno o más productos a un consumidor o usuario. Este conjunto de recursos tiene inicio y fin definidos y generalmente operan de acuerdo a un plan. Tal plan es frecuentemente documentado y especifica el producto a ser entregado o implementado, los recursos y fondos usados, el trabajo a ser realizado y el calendario para realizar dicho trabajo. [CMMI]

### 2. Cantidad de requerimientos

El número detallado de servicios y restricciones del sistema. [SOM 2002]

### 3. Requerimientos funcionales

Son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares y de cómo se comportará en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. [SOM 2002]

Describen las funciones que el software ejecutará; por ejemplo, formatear algún texto o modular una señal. Algunas veces son conocidos como capacidades. [SWEBOK 2005]

### 4. Requerimientos no funcionales

Son aquellos que actúan como restricciones de la solución. Los requerimientos no funcionales a veces son conocidos como restricciones o requerimientos de calidad. [SWEBOK 2005]

### 5. Aplicación de software

Software diseñado para satisfacer las necesidades de un usuario. [CIS 2005]

### 6. Cliente

La persona u organización que paga por el producto y usualmente (pero no necesariamente) decide los requerimientos. [IEEE 830]

La persona u organización que desea usar el servicio, producto o resultado del proyecto. [PMBOK 2005]

### 7. Usuario final

Persona o personas, que operan o interactúan directamente con el sistema. [ISO 8402]

### 8. Satisfacción del usuario

La percepción del usuario de haber utilizado correctamente su tiempo, habiendo recibido respecto de sus propias expectativas y en un determinado contexto ambiental la mejor prestación posible del servicio o producto adquirido. [ADOC 2000]

### 9. Fecha de inicio del proyecto

La fecha (día, mes y año) cuando el cliente ha autorizado a la empresa que proceda a la realización del proyecto de software. [RDC 2006]

### 10. Fecha de terminación del proyecto

La fecha (día, mes y año) cuando la empresa de software ha realizado la entrega formal del producto de software que ha desarrollado. [RDC 2006]

### 11. Estimación

Un cálculo o evaluación tentativa del esfuerzo, duración, y/o costo aproximado para realizar un trabajo. [PMITBC 2005]

### 12. Personal

Conjunto de los empleados de una empresa. [RDC 2006]

### 13. Tamaño del personal

El número de personal que forman parte del equipo asignado al desarrollo del proyecto de software. [RDC 2006]

### 14. Costo

El valor monetario o precio de una actividad o componente del proyecto que incluye el valor monetario de los recursos requeridos para desarrollar y completar la actividad o componente, o producir el componente. [PMBOK 2005]

Es la cantidad de dinero que la empresa desarrolladora de software invierte en la realización de un proyecto informático. Es diferente del precio de compra, que es el valor que el cliente cancela por adquirir el producto que la empresa de software desarrolla. [RDC 2006]

### 15. Complejidad

El grado en el cual el diseño o implementación de un sistema es difícil de entender y verificar. [SEG 2003]

Compuesta de dos dimensiones: Complejidad del Negocio y Complejidad Técnica. Cada dimensión puede ser caracterizada por un conjunto de atributos. Los atributos de negocio típicos incluyen tamaño, ubicación y riesgo financiero. Los atributos técnicos típicos incluyen el nivel de integración de la tecnología, necesidades de seguridad, estabilidad del hardware/software y la experiencia del personal. [PMITBC 2005]

#### **16. Rol**

Una definición del comportamiento y responsabilidades de un individuo o conjunto de individuos que trabajan juntos como un equipo, dentro del contexto de una organización de ingeniería de software. [RUP]

#### **17. Experiencia del personal**

Generalmente se refiere al conocimiento de como llevar a cabo una actividad o una tarea específica, en lugar del conocimiento factual. [WIKI]

#### **18. Rotación**

Es la cantidad de personas que ingresan y que se desvinculan de la empresa durante el desarrollo del proyecto. [RDC 2006]

#### **19. Interrupciones**

Periodos de tiempo durante el cual el personal asignado al proyecto interrumpió las labores del mismo para realizar otras actividades [RDC].

#### **20. Planificación**

Esta actividad abarca tanto la planificación de alto nivel del proyecto y el análisis de requerimientos preliminar. Este se enfoca en los objetivos, stakeholders, riesgos, presupuestos y calendarios. [IFPUG 2005]

#### **21. Especificación**

Actividad estructurada durante la cual el equipo de desarrollo trabaja con el cliente para identificar y documentar que funcionalidad, que interfaces, que calidad es requerida. [IFPUG 2005]

#### **22. Diseño**

Actividad durante la cual el equipo de desarrollo crea un diseño general de alto nivel, después posiblemente un diseño detallado. Esta actividad con frecuencia se traslapa con la especificación y la construcción. [IFPUG 2005]

#### **23. Construcción**

El equipo de desarrollo realiza la programación y pruebas de unidades que produce nuevo software o cambia el software existente. [IFPUG 2005]

El término construcción de software se refiere a la creación detallada de software, a través de una combinación de codificación, verificación, pruebas de unidades, pruebas de integración y depuración. [SWEBOK 2005]

#### **24. Pruebas**

El planear y desarrollar los diferentes niveles de pruebas sobre el software, realizado por personal que puede ser independiente de los desarrolladores. Esta actividad frecuentemente se traslapa con la actividad de construcción. [IFPUG 2005]

Es una actividad en la cual un sistema, producto o componente es usado bajo condiciones definidas, los resultados son observados o almacenados y una evaluación es realizada para ver si cumplen adecuadamente algunos o todos los requerimientos. [EIA 731-1]

#### **25. Instalación**

La actividad de prepararse para la instalación del software que será utilizado por los usuarios finales, a continuación trabajar con ellos para realizar la instalación, la documentación para el usuario y su capacitación en el uso adecuado del software. [IFPUG 2005]

El proceso de definir la arquitectura, componentes, interfaces y otras características de un sistema o componente. [IEEE 610]

#### **26. Tarea**

La unidad más pequeña de trabajo sujeta a la responsabilidad de la gerencia, la cual puede incluir una o más acciones. Una tarea es una asignación bien definida de trabajo para una o más personas. [IEEE 1058]

#### **27. Volumen**

El número de páginas generadas por documento desarrollado. [RDC 2006]

#### **28. Documentación**

La descripción narrativa y gráfica de un sistema. La documentación para un sistema de información incluye la documentación del sistema, documentación técnica, y sus procedimientos operativos. [PMITBC 2005]

**29. Documento**

Un medio y la información registrada en el, que generalmente tiene permanencia y puede ser leído por una persona o una máquina. Ejemplos incluyen los planes de administración de proyectos, especificaciones, procedimientos, estudios y manuales. [PMBOK 2005]

**30. Esfuerzo**

El número de horas requeridas para completar una tarea, un entregable, una fase o un proyecto. El enfoque aquí es sobre la cantidad de tiempo expresado en horas. [PMITBC 2005]

**31. Defecto**

Resultado no deseado de las tareas realizadas en el proyecto que implican 1) volver a realizar las mismas, 2) retrasar la fecha de entrega del proyecto, y/o 3) el malfuncionamiento del software. [RDC 2006]

Un problema en la especificación, diseño o implementación de un producto. [PSM 92]

**32. Falla**

La interrupción inesperada de la capacidad de un sistema para desarrollar sus funciones requeridas. [SEG 2003]

**33. Inspecciones**

Una actividad semejante a medir, examinar, probar o estimar una o más características de una entidad y comparar los resultados con los requerimientos especificados para establecer si la conformidad está siendo alcanzada en cada característica. [ISO 8402]

**34. Código Fuente**

El código fuente es un conjunto de líneas de código que conforman un bloque de texto que normalmente genera otro código mediante un compilador o intérprete para ser ejecutado por una computadora. Normalmente se refiere a la programación de software. Un único programador o un equipo de ellos escriben el código fuente en el lenguaje de programación elegido. [WIKI]

**35. Módulo**

Una parte lógicamente separable de un programa. [IEEE 610]

Los módulos son usualmente compilados separadamente y proveen un mecanismos de abstracción u ocultamiento de información a fin de que una implementación del modulo pueda ser

cambiada sin requerir ningún cambio a otros módulos. [SEG 2003]

**36. Complejidad ciclométrica**

Es una métrica del software que proporciona una medición cuantitativa de la complejidad de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

**37. Tiempo en corregir defectos**

El tiempo requerido expresado en horas para resolver un problema existente en el software. [RDC 2006]

**Fuentes**

ADOC 2000	Anales de documentación N° 3, 2000. La satisfacción del usuario un concepto en alza.
CIS 2005	Compendio de Ingeniería de software Juan Palacio
CMMI	The Capability Maturity Model – Continuous Representation, CMU/SEI 2000-TR-019,2000
EIA 731-1	System Engineering Capability Model, Electronic Industries Alliance, 2002
IEEE 610	Software Engineering Terms, IEEE, 1990
IEEE 830	Guide to Requirements Specification, IEEE, 1984
IEEE 610	Software Engineering Terms, IEEE, 1990
IFPUG 2005	Data collection questionnaire ,Sized Using IFPUG or NESMA FUNCTION POINTS IFPUG 2005
ISO 8402	Quality Management and Quality Assurance, International Organization for Standardization, 1994
JAMES 93	James Martin. Principles of object-oriented analysis and design, Prentice-Hall, 93
MICROSOFT	Microsoft TechNet, <a href="http://www.microsoft.com/technet/itsolutions/msf/default.msp">http://www.microsoft.com/technet/itsolutions/msf/default.msp</a>
PMBOK 2005	A guide to the Project Management Body of Knowledge (PMBOK Guide) American National Standard 2004
PMITBC 2005	Project Management for Information, Technology, Business, and Certification, Gopal K. Kapur, 2005
PRESSMAN 98	Ingeniería del software, Pressman Roger, 1998
PSM 92	Practical Software Metrics for Process Management and Process Improvement, Grady Robert 1992.
RDC 2006	Referencia Definida a Conveniencia, VLIR-8 SUB-Componente de Ingeniería de Software, 2006
SEG 2003	Software Engineering Glossary, SYNSPACE, 2003
SOM 2002	Ingeniería de Software 6ta Edición, Somerville Ian, 2002
SWEBOK 2005	Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE, 2004
SUTTON 2001	Universidad de Belgrano <a href="http://www.monografias.com/trabajos16/capacitacion-personal/capacitacion-personal.shtml">http://www.monografias.com/trabajos16/capacitacion-personal/capacitacion-personal.shtml</a>
WIKI	WIKIPEDIA La enciclopedia libre, <a href="http://es.wikipedia.org">http://es.wikipedia.org</a>

**REFERENCIA PARA EL INGRESO DE VALORES VALIDOS (GR)**

<b>Cod.</b>	<b>DATOS</b>	<b>REFERENCIAS</b>
1	Proyecto	Ingresar el nombre o código que identifica al proyecto dentro de la empresa de software
2	Requerimientos	Corresponde al número total de requerimientos que se acuerdan entregar entre el cliente y la empresa al firmar un contrato legal al inicio del proyecto
3	Aplicación de software	Refleja la actividad productiva a la que esta dirigido el producto de software desarrollado, Ej.: Aplicación financiera, contable, Workflow, administrativa, Portal Web, etc.
4	Proyecto de software	Especificar si se trata de los siguientes casos: 1) Nuevo desarrollo de software. 2) La mejora de uno existente o 3) Adaptación del software por cambios en la tecnología usada
5	Satisfacción del cliente	El valor numérico que se obtiene de aplicar una encuesta de satisfacción al cliente. Cada empresa debe definir estas encuestas tomando en cuenta los factores más importantes según el criterio de cada una.
6	Satisfacción del usuario	Situación similar a la satisfacción del cliente pero aplicada al usuario final del software
7	Fecha inicio	Corresponde a la fecha en la que empezó oficialmente el proyecto de software
8	Fecha fin	La fecha en la que se hizo la entrega formal del software desarrollado al cliente
9	Personal estimado	La cantidad de personas que son asignadas al proyecto al inicio del proyecto
10	Personal real	La cantidad total de personas que realmente participaron en el proyecto de software contabilizados al término del mismo.
11	Costo estimado	El costo o cotización calculada al inicio del proyecto de software, expresado en dólares.
12	Costo real	El costo final que tuvo el proyecto, contabilizado al término de mismo y expresado en dólares.
13	Complejidad negocio	El valor promedio obtenido de la aplicación de encuesta que evalúa atributos clave que afectan el tipo de negocio del cliente. Se evalúa al inicio del proyecto de software.
14	Complejidad técnica	El valor promedio obtenido de la aplicación de encuesta que evalúa atributos clave que afectan el tipo de proyecto de software que desarrollará la empresa. Se evalúa al inicio del proyecto de software.
15	Rol	Representa las funciones que desempeña una persona durante el desarrollo del proyecto: Ej. Líder de proyecto, Programador, Arquitecto de SW, Cliente, etc.
16	Nombre	Nombre y Apellido de cada una de las personas asignadas al proyecto
17	Experiencia desarrollo	La cantidad de meses que lleva una persona desarrollando proyectos de software
18	Experiencia negocio	La cantidad de años que las personas sobre el área de negocio del cliente
19	Rotación	Para cada una de las personas, debe darse solo uno de los siguientes tres casos: 1) Permaneció el tiempo requerido, 2) No permaneció el tiempo requerido y 3) Fue reemplazado por otra persona.
20	Tiempo	En este campo se registra el tiempo que duró una interrupción de las labores de un proyecto de software. Este valor se ingresa

		cuando la interrupción ha cesado.
21	Causa	Se escribe una descripción breve de la causa que motivó la suspensión del desarrollo del proyecto.
22	Fase	Corresponde a las macro-actividades de un proyecto como la Planificación, Especificación de requerimientos, Diseño, Construcción, Pruebas e Instalación. Pero que son propias de cada modelo o metodología que se utilice en la empresa.
23	Número	Es un número secuencial que identifica cada una de las tareas a desarrollarse en el proyecto.
24	Tarea / Entregable	El nombre o código que identifica la tarea que se está definiendo.
25	Responsable	El nombre de la persona que responde por el cumplimiento de la tarea. Solo puede haber un responsable por tarea definida.
26	Páginas	Se registra el número de páginas que tiene el documento una vez que ha sido terminado
27	Documentación	Se ingresa el nombre del documento ha elaborarse. Ej. Presupuesto, Cronograma, Lista de requerimientos, Reporte de reunión, etc.
28	Descripción	Se da una breve descripción de la tarea ha ser desarrollada. Se utiliza esta información como contexto de los datos cuantitativos que se ingresan.
29	Criterio terminación	El criterio que se define para saber si una tarea esta terminada.
30	Fecha estimada inicio	La fecha planificada en la que debería empezar la tarea
31	Fecha estimada fin	La fecha planificada en la que debería terminar la tarea
32	Fecha real inicio	La fecha real en la que se empezó a desarrollar la tarea
33	Fecha real fin	La fecha real en la que se terminó la tarea, habiendo cumplido con el criterio de terminación definido.
34	Duración estimada	La cantidad de tiempo que debería durar esta tarea, expresada en días. El valor más pequeño para este campo es 1 día.
35	Duración real	La cantidad de tiempo que duró esta tarea, expresada en días. El valor más pequeño para este campo es 1 día.
36	Costo estimado	El costo que se presupuesta para esta tarea, expresada en dólares.
37	Costo real	El costo que se contabiliza para esta tarea, expresada en dólares.
38	Esfuerzo estimado	La cantidad acordada de horas que cada una de las personas debe emplear para desarrollar la tarea antes de realizarla.
39	Esfuerzo real	La cantidad contabilizada de horas que cada una de las personas asignadas a esta tarea tuvo que emplear para terminarla.
40	Número de reporte	Es un número secuencial que identifica cada uno de los reportes de problemas o cambios en requerimientos ocurridos durante el desarrollo del proyecto y durante los 3 primeros meses de uso formal por parte del cliente.
41	Fecha	Es la fecha en la que se realiza este reporte que puede no coincidir con la fecha en la que se detectó el problema que se informa.
42	Estado	Se marca solo uno de cuatro estados diferentes en los que puede encontrarse un problema. Es decir: 1) Al momento de realizar este reporte se debe marcar el estado "en cola". 2) En el momento que este problema empieza a ser resuelto debe marcarse el estado "Abierto" además de dejar registrada la fecha en la que se lo empieza a resolver. 3) Al resolver satisfactoriamente este problema se debe marcar el estado "Cerrado". Y finalmente, 4) Si un problema que se ha reportado no va ha ser tratado por el motivo que fuese se debe marcar el estado "Descartado".

43	Fecha abierto	La fecha en la que un problema que ha sido previamente reportado empieza a ser tratado por las personas asignadas a esta tarea.
44	Tipo	Se distinguen los diferentes problemas que surgen durante el desarrollo del proyecto entre defectos y fallas.
45	Severidad	Se debe escoger una severidad del problema reportado según el criterio que se utilice para calificar cada uno de estos problemas. En este caso puede ser "leve", "grave" o "crítica".
46	Esfuerzo estimado	La cantidad aproximada de horas-hombre que se piensa se necesitará para resolver un problema.
47	Esfuerzo real	La cantidad total de horas-hombre que se utilizó realmente para resolver el problema.
48	Requerimiento	En caso de reportar un cambio sobre los requerimientos definidos al inicio del proyecto de software. Se debe escoger uno de los siguientes tres casos: 1) "Agregado", es decir, se aumenta la funcionalidad del producto. 2) "Modificado", es decir, se altera uno de los requerimientos previamente definidos y 3) "Eliminado", es decir, que no se desarrollo o se deja de desarrollar uno de los requerimientos del software acordados con el cliente.
49	Funcional / No funcional	Si el requerimiento tiene que ver con la funcionalidad o tareas que debe cumplir el software se marcará "Funcional". Caso contrario se marcará "No Funcional".
50	Método que detectó el problema	Se deberá escoger de entre las opciones listadas el método que se utilizó y a través del cual se detecto el problema.
51	Observaciones	Se proporcionará una breve descripción acerca del método que detectó el problema en caso de ameritarlo.
52	Descripción del problema	Una descripción detallada del problema reportado.
53	Componente afectado	Indicar a que proyecto, sistema, modulo o clase pertenece o afecta el problema.
54	Causas	Un listado de las posibles causas que generaron este inconveniente
55	Efectos	Listar los efectos que tuvo el problema sobre el proyecto.
56	Fecha Mes	Ingresar el mes en el que se realiza la actividad de programación del software.
57	Líneas de código	El número de líneas de código que son escritas por cada uno de los programadores
58	Componente, método o procedimiento	Hacer el listado de los diferentes componentes que conforman la aplicación de software desarrollada.
59	Horas de desarrollo del componente	La cantidad total de esfuerzo que se utilizó para desarrollar uno de los componentes del software
60	Número de defectos	La cantidad total de defectos en los módulos del software que fueron detectados durante la fase de pruebas
61	Complejidad ciclomática	Este valor es calculado para los métodos y procedimientos que conforman el código fuente.
62	Demora de aprendizaje	Las cantidad de horas que le toma al personal nuevo familiarizarse con los componentes de software ya desarrollados

## REFERENCIAS

- [1] Sommerville, IAN. Ingeniería de software: 6ta edición. Addison-Wesley. 2002.
- [2] Roger Pressman. Ingeniería del software, un enfoque practico. 4ta Edición. McGraw Hill. 1998.
- [3] Corporación de Promoción de Exportaciones e Inversiones del Ecuador (CORPEI). Estudio del Sector de Tecnología en Sistemas de Información de Ecuador: Software y Servicios. Ecuador, 2001.  
<http://www.ecuadorexporta.org/>
- [4] ESPOL-VLIR, SUB-Componente 8 Ingeniería de software. Estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en Quito, Guayaquil y Cuenca. Danny Salazar, Mónica Villavicencio, Monique Snoeck, Verónica Macías. Ecuador, 2003.  
<http://www.jornadasis.espol.edu.ec/ljornadas>
- [5] Asociación Ecuatoriana de Software. Primer estudio de la industria del software del Ecuador. Ecuador, 2005. <http://www.aesoft.com>



- [6] GST Latinoamérica. Comparación de las iniciativas latinoamericanas para mejorar la producción de software. Dr. Marcelo Jenkins. San Pedro-Costa Rica. 2005.
- [7] Software Engineering Institute, Carnegie Mellon University. Goal-driven software measurement - A guidebook. Robert E. Park, Wolfhart B. Goethert, William A. Florac. EEUU – 1996.  
[www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html](http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html)
- [8] Stephen H. Kan, Foreword by Caper Jones. Metrics and models in software. Second Edition. Addison – Wesley 2003.
- [9] Dale H. Besterfield, PhD. Control de Calidad. 4ta. Edición. PrenticeHall Hispanoamérica. México – 1995.
- [10] Robert B. Grady. Practical Software metrics for project management and process improvement. Hewlett – Packard 1992.
- [11] Dr. Marcelo Jenkins C. CMMI V1.1 Tutorial. GST Latinoamérica. San Pedro – Costa Rica 2005.

[12] Software Metrics Guide.

[http://sunset.usc.edu/classes/cs577b\\_2001/metricsguide/metrics.html](http://sunset.usc.edu/classes/cs577b_2001/metricsguide/metrics.html).

Ultima visita: Julio 2005.

[13] Cem Kaner, Walter P. Bond. International Software Metrics Symposium, Metrics 2004. Software Engineering Metrics: What do they measure and how do we know? Julio 2004.

<http://www.kaner.com/pdfs/metrics2004.pdf>

[14] IEEE Computer Society. Guide to the software engineering body of knowledge. 2004. <http://www.swebok.org/>

[15] John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean. Practical Software Measurement, Objective Information for decision makers Fred Hall Addison-Wesley - 2002.

[16] Microsoft Corporation. Microsoft Solution Framework, white paper. 2003. <http://www.microsoft.com/msf>

[17] Suzanne Robertson, James Robertson. Mastering the requirements process. Second edition. Addison-Wesley - 2006.

- [18] William A. Florac. Anita D. Carleton. Measuring the software process, statistical process control for software process improvement. Addison-Wesley - 1999.
- [19] Roberto Hernández Sampieri, Carlos Fernandez Collado, Pilar Baptista Lucio. Metodología de la investigación. McGraw HILL - 1991.
- [20] Stephen H. Kan, Foreword by Caper Jones. Metrics and models in software quality engineering. Second edition. Addison-Wesley – 2003.
- [21] Paul Goodman. Software metrics, best practices for successful IT management. Rosthstein Associates Inc., Publisher - 2004.
- [22] Fausto Chicango. I Jornadas de Ingeniería de software. Transición de las empresas para pasar de un esquema de desarrollo en cascada a iterativo utilizando RUP. Centro de publicación y difusiones ESPOL - 2004.
- [23] Hitoshi Kume. Herramientas estadísticas básicas para el mejoramiento de calidad. Grupo editorial Norma. 2002.

- [24] Gopal K. Kapur. Center for project management. Project management for information, technology, business, and certification. Pearson Prentice Hall - 2005.
- [25] Grupo SETI S.A. de CV. Midiendo la Complejidad del Software. Ing. José Manuel Milanés.  
[www.gruposeti.com/Recursos/PDFs/COMPLEJIDAD.pdf](http://www.gruposeti.com/Recursos/PDFs/COMPLEJIDAD.pdf)
- [26] Monografías .COM. Recolección de datos. Jose A Avilez M.  
<http://www.monografias.com/trabajos12/recoldat/recoldat.shtml#quees>
- [27] Daniel Chudnovsky, Andrés López, Silvana Meltisko. El sector del software y servicios informáticos (SSI) en Argentina: Situación actual y perspectivas de desarrollo. Argentina, 2001. [www.fund-cenit.org.ar/Descargas/dt27.pdf](http://www.fund-cenit.org.ar/Descargas/dt27.pdf). Última visita: Julio 2004.
- [28] CMMI. CMU/SEI 2000-TR-O-19. The Capability Maturity Model – Continuous Representation 2000.
- [29] IEEE 830. Standard Guide to Requirements Specification. IEEE, 1990.

- [30] PMBOK 2005. A guide to the Project Management Body of Knowledge (PMBOK Guide) American National Standard 2004.
- [31] ADOC 2000. ISO 8402. Quality Management and Quality Assurance, International Organization for Standardization, 1994.
- [32] Anales de documentación N° 3, 2000. La satisfacción del usuario un concepto en alza.
- [33] RDC 2006. Referencia Definida a Conveniencia, VLIR-8 SUB-Componente de Ingeniería de Software, 2006.
- [34] Gipsy Bracco. Elaboración y análisis de métricas para el proceso de desarrollo de software en empresas grandes desarrolladoras de software del Ecuador. Tesis de grado ESPOL 2006.