

TESIS DEFENDIDA POR
Julio César Fonseca Caballero
Y APROBADA POR EL SIGUIENTE COMITÉ

M.C. Josefina Rodríguez Jacobo

Director del Comité

Dr. Jesús Favela Vara

Miembro del Comité

Dra. Carmen Guadalupe Paniagua
Chávez

Miembro del Comité

Dr. Pedro Gilberto López Mariscal

*Coordinador del programa de
Ciencias de la Computación*

Dr. Raúl Ramón Castro Escamilla

Director de Estudios de Postgrado

22 de Septiembre del 2006

**CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN
SUPERIOR DE ENSENADA**



PROGRAMA DE POSGRADO EN CIENCIAS DE LA COMPUTACIÓN

**SISTEMA DE APOYO AL PROCESO DE MEDICIÓN EN EQUIPOS DE
TRABAJO DISTRIBUIDOS**

TESIS

Que para cubrir parcialmente los requisitos necesarios para obtener el grado de
MAESTRO EN CIENCIAS

Presenta:

JULIO CÉSAR FONSECA CABALLERO

Ensenada, Baja California, México. Septiembre del 2006.

RESUMEN de la tesis de **JULIO CÉSAR FONSECA CABALLERO**, presentada como requisito parcial, para la obtención del grado de **MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN**. Ensenada, Baja California, México. Agosto del 2006.

SISTEMA DE APOYO AL PROCESO DE MEDICIÓN EN EQUIPOS DE TRABAJO DISTRIBUIDOS

Resumen aprobado por:

M.C. Josefina Rodríguez Jacobo
Directora de Tesis

Las métricas del software son entidades de información que se presentan en cualquier desarrollo de software y constituye información valiosa que puede aplicarse para mejorar el desarrollo de productos de software, su aplicación puede ir desde monitorear el trabajo de un desarrollador hasta cuantificar el desempeño de toda una organización.

El trabajar en equipos de desarrollo de software es una práctica cada vez más común por lo que resulta muy útil llevar un control de estos a través de métricas de software. Sin embargo, este control ha resultado deficiente principalmente porque solo se manejan las métricas desde una perspectiva de hacer una estimación y calcular el valor actual que se obtiene y posiblemente revisar alguna gráfica. Para obtener información más completa que permita tomar decisiones sobre el proyecto, proceso de desarrollo, producto y los recursos es necesario implementar un proceso de medición detallado.

En este trabajo se propone la guía SARE para la implantación de un proceso de medición adaptado a equipos de trabajo y bajo un contexto administrativo donde se aplican las métricas y se promueve la participación de los miembros del equipo dentro del proceso de medición.

Tomando como base la guía SARE, se implementó una herramienta para apoyar la comprensión de los lineamientos dados por la guía.

Palabras clave: Métricas de software, ingeniería de software, desarrollo en equipo, administrador de proyectos.

ABSTRACT of the thesis presented by JULIO CÉSAR FONSECA CABALLERO, as a partial requirement to obtain the MASTER IN SCIENCES degree in COMPUTER SCIENCES. Ensenada, Baja California, México. September 2006

MEASUREMENT PROCESS SUPPORT TOOL FOR DISTRIBUTED SOFTWARE TEAMS

Abstract approved by:

M.Sc. Josefina Rodríguez Jacobo
Thesis Director

Software metrics are information entities that are present in any software development, these metrics are valuable information that can be applied to improve software development processes, and their application goes from monitoring a programmer performance to obtaining the productivity for an entire corporation.

Nowadays, developing systems through software development teams is a common practice. Thus, it is useful to control those teams with software metrics. Even though this control has not been good enough, mainly because there is a metric management just for estimations, for getting the earned value and maybe reviewing a chart, in order to take better decisions about project, processes, products and resources is necessary to implement a detailed software measurement process.

The purpose of this research is to propose a process called SARE which is adapted for the work in teams. Also is defined a management context for the metrics application; this context also promotes team member participation over the measurement process.

Based on the SARE guide, a tool was developed; this SARE tool supports the guide.

Keywords: Software metrics, software engineering, team development, project management.

Dedicatorias

A mi mamá y familia del Oso: Ma, Oso, Clau y Mariana,
Por situarme siempre en una realidad y ver en ellos el valor para enfrentar la vida.

A mi familia,
Tía Fega, Tío Toño, Yola, Tía Guille, Denisse, Pepe, Abuelo y ¡todos!
Por empujarme a cumplir este objetivo y fortalecerme con su ánimo.

A mi novia: Vanessita
Por estar ahí, creer en mí y motivarme a alcanzar mis metas.

A mis amigos cicesianos: Omar, Alfredo, Luis, Plebes, Chris, Angelopolus,
Carmen, ¡todos!
Les deseo siempre lo mejor, los llevo en mi corazón a cada uno.

A mis amigos: de México, de Ensenada, de Dinamarca, de Rusia
Por tener siempre buenos deseos para que yo alcanzara esta meta.

Y a la familia de mi novia: Doña Lupe, Don Nato (qdp), Maritza y Midori
Gracias por abrirme las puertas de su casa y darme la confianza de platicarles
todos mis rollos.

Agradecimientos

A Dios,
Señor que gran aventura me tenías preparada, solo tu sabías que sería para mí
una gran lección de vida.

A mi asesora la Dra. Josefina Rodríguez Jacobo,
Por confiar en mí y ayudarme a cumplir mi sueño de explorar las métricas del
software.

Al Dr. Jesús Favela Vara y al Dra. Carmen Guadalupe Paniagua Chávez,
por su paciencia, sinceridad y aportaciones a este trabajo de tesis.

A todo el personal del Departamento de Ciencias de la Computación,
por todo el entusiasmo que se refleja en el programa y que me motivo a cumplir
mis metas.

Al Centro de Investigación Científica y de Educación Superior de Ensenada
(CICESE), por permitirme echar un vistazo por la ventana de la investigación y por
el apoyo económico.

Al Consejo Nacional de Ciencia y Tecnología (Conacyt),
por su ayuda para mi manutención.

Al ambiente que me ha dado Ensenada
ya casi siete años de conocer gente agradable, también buenas horas de tenis y
volviéndome loco por recorrer cada vez mas esta bella península

CONTENIDO

	Página
CAPÍTULO I. INTRODUCCIÓN.....	1
I.1. INTRODUCCIÓN	1
I.2. PLANTEAMIENTO DEL PROBLEMA.....	3
I.3. OBJETIVO DEL TRABAJO DE TESIS.....	3
I.4. ORGANIZACIÓN DE LA TESIS	4
CAPÍTULO II. FUNDAMENTOS	5
II.1. INTRODUCCIÓN.....	5
II.2. MÉTRICA, MEDICIÓN E INDICADORES	5
II.3. TIPOS DE MÉTRICAS	6
<i>II.3.1. Métricas del proceso y del producto</i>	<i>7</i>
<i>II.3.2. Métricas objetivas y subjetivas</i>	<i>7</i>
<i>II.3.3. Métricas directas e indirectas.....</i>	<i>7</i>
<i>II.3.4. Ciclo de vida</i>	<i>8</i>
<i>II.3.5. Herramienta</i>	<i>8</i>
II.4. PROCESO DE MEDICIÓN (PM)	8
<i>II.4.1. Formulación.....</i>	<i>8</i>
<i>II.4.2. Recolección</i>	<i>9</i>
<i>II.4.3. Análisis.....</i>	<i>9</i>
<i>II.4.4. Interpretación.....</i>	<i>10</i>
<i>II.4.5. Retroalimentación.....</i>	<i>11</i>
II.5. PLAN DE MEDICIÓN	11
II.6. MÉTRICAS DE LA ADMINISTRACIÓN DE PROYECTOS.....	12
<i>II.6.1. Tamaño.....</i>	<i>13</i>
<i>II.6.2. Esfuerzo.....</i>	<i>13</i>
<i>II.6.3. Calidad.....</i>	<i>13</i>
<i>II.6.4. Duración</i>	<i>13</i>

CONTENIDO (CONTINUACIÓN)

	Página
II.7. MARCOS DE TRABAJO EXISTENTE.....	13
<i>II.7.1. PSM</i>	<i>13</i>
<i>II.7.2. TSP</i>	<i>14</i>
II.8. HERRAMIENTAS.....	15
<i>II.8.1. Fases del proceso de medición</i>	<i>16</i>
<i>II.8.2. Productos del proceso de desarrollo</i>	<i>18</i>
<i>II.8.3. Administración de proyectos.....</i>	<i>20</i>
<i>II.8.4. Utilidades de apoyo.....</i>	<i>21</i>
II.9. HERRAMIENTAS PRINCIPALES	21
<i>II.9.1. DataDrill.....</i>	<i>22</i>
<i>II.9.2. PSM Insight Version 4.2.2</i>	<i>22</i>
<i>II.9.3. TSPi Guide</i>	<i>22</i>
CAPÍTULO III. PROCESO DE MEDICIÓN PARA EQUIPOS DE TRABAJO DISTRIBUIDOS	25
III.1. INTRODUCCIÓN	25
III.2. PRÁCTICAS ENFOCADAS A LA GUÍA	27
<i>III.2.1. Proceso de medición para equipos distribuidos.....</i>	<i>27</i>
<i>III.2.2. Consideraciones de TSPi.....</i>	<i>28</i>
<i>III.2.3. Equipos de trabajo.....</i>	<i>29</i>
<i>III.2.4. Métricas para los elementos de la IS.....</i>	<i>30</i>
III.3. COMPONENTES DE INFORMACIÓN.....	37
III.4. PRINCIPIOS.....	37
CAPÍTULO IV. ANÁLISIS Y DISEÑO DEL SISTEMA SARE	39
IV.1. INTRODUCCIÓN	39
IV.2. ANÁLISIS.....	39
<i>IV.2.1. Requerimientos.....</i>	<i>39</i>
<i>IV.2.2. Casos de uso.....</i>	<i>49</i>

CONTENIDO (CONTINUACIÓN)

	Página
IV.2.3. Modelo dinámico (Diagramas de secuencia)	54
IV.3. DISEÑO	62
IV.3.1. Arquitectura	62
IV.3.2. Subsistemas	64
IV.3.3. Diagramas de clases	72
CAPÍTULO V. IMPLEMENTACIÓN Y FUNCIONALIDAD DEL SISTEMA SARE	78
V.1. INTRODUCCIÓN	78
V.2. IMPLEMENTACIÓN DEL SISTEMA SARE	78
V.2.1. Lenguaje de programación	78
V.2.2. Base de datos	79
V.2.3. Servidor de correo	79
V.3. FUNCIONALIDAD DEL SISTEMA SARE	79
V.3.1. Acceso al sistema SARE	81
V.3.2. Administración	82
V.3.3. Formulación de mediciones	89
V.3.4. Recolección	91
V.3.5. Revisar monitoreos	98
V.3.6. Interpretar monitoreo	102
V.3.7. Revisión retroalimentación	105
V.3.8. Utilerías	108
CAPÍTULO VI. PRUEBAS AL SISTEMA SARE	111
VI.1. INTRODUCCIÓN	111
VI.2. DISEÑO DE LAS PRUEBAS	111
VI.3. PRUEBAS DE ACEPTACIÓN	111
VI.3.1. Objetivos y factores de motivación de las pruebas	112
VI.3.2. Metodología	112

CONTENIDO (CONTINUACIÓN)

	Página
VI.3.3. Resultados	126
VI.4. DISCUSIÓN DE LOS RESULTADOS DE LAS PRUEBAS.....	128
VI.4.1. Pruebas de aceptación.....	129
CAPÍTULO VII. CONCLUSIONES, APORTACIONES Y TRABAJO FUTURO ..	130
VII.1. CONCLUSIONES.....	130
VII.2. APORTACIONES	130
VII.3. TRABAJO A FUTURO.....	131
BIBLIOGRAFÍA	132
APÉNDICE A. ANÁLISIS Y DISEÑO DE SARE	134
A.1 CASOS DE USO	134
A.1.1 Formular métrica	134
A.1.2 Participar en retroalimentación.....	135
A.1.3 Configurar proyecto.....	136
A.1.4 Administrar equipos de trabajo.....	136
A.1.5 Administrar requerimientos	137
A.1.6 Administrar módulos	138
A.1.7 Administrar pruebas.....	138
A.1.8 Discutir en grupo.....	139
A.1.9 Notificar.....	140
A.2 DIAGRAMAS DE CLASES.....	140
A.2.1 Conexión a la base de datos.....	141
A.2.2 Notificaciones.....	141
A.2.3 Formulación de métricas.....	143
A.2.4 Recolección	143
A.2.5 Discusiones grupales.....	150

CONTENIDO (CONTINUACIÓN)

	Página
<i>A.2.6 Elementos genéricos interfaz gráfica</i>	151
<i>A.2.7 Monitoreos, interpretación y retroalimentación</i>	151
<i>A.2.8 Componentes genéricos</i>	156
<i>A.2.9 Contexto administrativo</i>	157
A.3 MODELO DE DATOS (TABLAS)	165
<i>A.3.1 Contexto administrativo</i>	165
<i>A.3.2 Formulación de mediciones</i>	170
<i>A.3.3 Recolección</i>	171
<i>A.3.4 Monitoreos, interpretación y retroalimentación</i>	173
APÉNDICE B. DATOS DE LAS PRUEBAS	178
B.1 ALTA DEL PROYECTO	178
B.2 ROLES Y EQUIPOS DE TRABAJO	179
B.3 CONTEXTO ADMINISTRATIVO	180
<i>B.3.1 Requerimientos</i>	180
<i>B.3.2 Diseño</i>	182
<i>B.3.3 Implementación</i>	182
<i>B.3.4 Casos de prueba</i>	183
B.4 FORMULACIÓN	183
B.5 RECOLECCIÓN	183
B.6 ANOMALÍAS	184
B.7 INTERPRETACIÓN	185
B.8 RETROALIMENTACIÓN	185
APÉNDICE C. MANUAL DE INSTALACIÓN DEL SISTEMA SARE	186

LISTA DE FIGURAS

	Página
<i>Figura 1. Relación entre los términos medición, métrica e indicador.....</i>	<i>6</i>
<i>Figura 2. Proceso de medición para equipos de trabajo distribuidos.....</i>	<i>26</i>
<i>Figura 3. Gráfica rica del proceso de medición orientado a equipos de trabajo.....</i>	<i>36</i>
<i>Figura 4. Diagrama del caso de uso Calcular métrica.....</i>	<i>50</i>
<i>Figura 5. Diagrama del caso de uso monitorear métrica.</i>	<i>52</i>
<i>Figura 6. Diagrama del caso de uso participar en interpretación.</i>	<i>53</i>
<i>Figura 7. Diagrama de secuencia estimar Métrica.....</i>	<i>55</i>
<i>Figura 8. Diagrama de secuencia calcular valor actual.....</i>	<i>57</i>
<i>Figura 9. Diagrama de secuencia revisar monitoreo.....</i>	<i>59</i>
<i>Figura 10. Diagrama de secuencia participar en Interpretación</i>	<i>61</i>
<i>Figura 11. Arquitectura a nivel físico de SARE.</i>	<i>63</i>
<i>Figura 12. Arquitectura lógica y estructura de capas de SARE.</i>	<i>64</i>
<i>Figura 13. Diagrama de Subsistemas.....</i>	<i>64</i>
<i>Figura 14a. Estimar métrica.</i>	<i>72</i>
<i>Figura 15. Calcular valor actual.....</i>	<i>75</i>
<i>Figura 16. Revisar Monitoreo</i>	<i>76</i>
<i>Figura 17. Participar en Interpretación.....</i>	<i>77</i>
<i>Figura 18. Plantilla genérica de SARE.</i>	<i>80</i>
<i>Figura 19. Pantalla de bienvenida.</i>	<i>81</i>
<i>Figura 20. Pantalla de administración.</i>	<i>83</i>
<i>Figura 21. Creación de un nuevo ingeniero de software.</i>	<i>84</i>
<i>Figura 22. Creación de un nuevo equipo.</i>	<i>84</i>
<i>Figura 23. Asignación de ingenieros a equipos.</i>	<i>85</i>
<i>Figura 24. Pantalla de visualización de equipos de trabajo.....</i>	<i>86</i>
<i>Figura 25. Listado de Requerimientos.</i>	<i>87</i>
<i>Figura 26. Listado de módulos.....</i>	<i>88</i>
<i>Figura 27. Relación entre módulos.</i>	<i>88</i>
<i>Figura 28. Relación de un módulo con submodulos.</i>	<i>89</i>

LISTA DE FIGURAS (CONTINUACIÓN)

	Página
<i>Figura 29. Requerimientos asociados.</i>	89
<i>Figura 30. Métricas del ciclo.</i>	90
<i>Figura 31. Pantalla para el alta de una métrica.</i>	91
<i>Figura 32. Pantalla principal de mediciones (recolección).</i>	92
<i>Figura 33. Listado de anomalías históricas.</i>	93
<i>Figura 34. Detalle de datos históricos.</i>	93
<i>Figura 35. Reestimaciones de las métricas.</i>	94
<i>Figura 36. Monitoreos de las métricas asociadas.</i>	95
<i>Figura 37. Calendario de actividades.</i>	96
<i>Figura 38. Listado de estimaciones.</i>	97
<i>Figura 39. Registro del valor actual.</i>	97
<i>Figura 40. Calcular valor actual.</i>	98
<i>Figura 41. Listado de Monitoreos.</i>	99
<i>Figura 42. Datos del monitoreo.</i>	100
<i>Figura 43. Monitoreos asociados.</i>	101
<i>Figura 44. Anomalías.</i>	102
<i>Figura 45. Botón para Registrar información.</i>	102
<i>Figura 46. Interpretación.</i>	103
<i>Figura 47. Información adicional.</i>	104
<i>Figura 48. Recomendaciones.</i>	104
<i>Figura 49. Convocatoria discusión.</i>	105
<i>Figura 50. Retroalimentación.</i>	106
<i>Figura 51. Métricas a monitorear.</i>	107
<i>Figura 52. Ruta de retroalimentación.</i>	108
<i>Figura 53. Foro de discusión.</i>	109
<i>Figura 54. Mi cuenta.</i>	110
<i>Figura 55. Diagrama general de casos de uso.</i>	134

LISTA DE TABLAS

	Página
<i>Tabla I. Comparación de las herramientas principales</i>	<i>22</i>
<i>Tabla II. Tabla de especificación de hardware y software.</i>	<i>79</i>
<i>Tabla III. Relación entre requerimientos funcionales y casos de prueba.</i>	<i>114</i>
<i>Tabla IV. Atributos y métodos de la clase ConexionBDBean.</i>	<i>141</i>
<i>Tabla V. Atributos y métodos de la clase en el servidor ObtenerCalendario.</i>	<i>141</i>
<i>Tabla VI. Atributos y métodos de la clase en el cliente CalendarioGrafico.</i>	<i>142</i>
<i>Tabla VII. Atributos y métodos de la clase en el servidor DatosActividad.</i>	<i>142</i>
<i>Tabla VIII. Atributos y métodos de la clase en el cliente DatosActividad.</i>	<i>142</i>
<i>Tabla IX. Atributos y métodos de la clase en el servidor Notificador.</i>	<i>142</i>
<i>Tabla X. Atributos y métodos de la clase en el servidor FormulacionMetricas.</i>	<i>143</i>
<i>Tabla XI. Atributos y métodos de la clase en el servidor ProcesarFormularioMetricas.</i>	<i>143</i>
<i>Tabla XII. Atributos y métodos de la clase en el cliente Analogia.</i>	<i>144</i>
<i>Tabla XIII. Atributos y métodos de la clase en el servidor Analogia.</i>	<i>144</i>
<i>Tabla XIV. Atributos y métodos de la clase en el cliente ParametrosModelo.</i>	<i>144</i>
<i>Tabla XV. Atributos y métodos de la clase en el servidor ParametrosModelo.</i>	<i>144</i>
<i>Tabla XVI. Atributos y métodos de la clase en el cliente DatosEstimacion.</i>	<i>145</i>
<i>Tabla XVII. Atributos y métodos de la clase en el cliente PrincipalRecoleccion.</i>	<i>145</i>
<i>Tabla XVIII. Atributos y métodos de la clase en el servidor PrincipalRecoleccion.</i>	<i>145</i>
<i>Tabla XIX. Atributos y métodos de la clase en el servidor ProcesarRecoleccion.</i>	<i>146</i>
<i>Tabla XX. Atributos y métodos de la clase en el servidor Modelos.</i>	<i>146</i>
<i>Tabla XXI. Atributos y métodos de la clase en el cliente DefinirModelo.</i>	<i>146</i>
<i>Tabla XXII. Atributos y métodos de la clase en el servidor ProcesarModelo.</i>	<i>147</i>
<i>Tabla XXIII. Atributos y métodos de la clase en el servidor VerEstimaciones.</i>	<i>147</i>
<i>Tabla XXIV. Atributos y métodos de la clase en el cliente ReestimacionMonitoreo.</i>	<i>147</i>
<i>Tabla XXV. Atributos y métodos de la clase en el servidor ObtenerEventosReestimacionMonitoreos.</i>	<i>148</i>
<i>Tabla XXVI. Atributos y métodos de la clase en el servidor ProcesarReestimacionMonitoreo.</i>	<i>149</i>

LISTA DE TABLAS (CONTINUACIÓN)

	Página
<i>Tabla XXVII. Atributos y métodos de la clase en el servidor ObtenerListadoEstimaciones.</i>	
.....	149
<i>Tabla XXVIII. Atributos y métodos de la clase en el cliente ListadoEstimaciones.</i>	149
<i>Tabla XXIX. Atributos y métodos de la clase en el cliente EstValAct.</i>	149
<i>Tabla XXX. Atributos y métodos de la clase en el servidor ObtenerRefMetricas.</i>	150
<i>Tabla XXXI. Atributos y métodos de la clase en el cliente WideBand</i>	150
<i>Tabla XXXII. Atributos y métodos de la clase en el servidor WideBand.</i>	151
<i>Tabla XXXIII. Atributos y métodos de la clase en el servidor ObtenerPlantillaElementosIS.</i>	
.....	151
<i>Tabla XXXIV. Atributos y métodos de la clase en el servidor ObtenerListadoMonitoreos.</i>	
.....	152
<i>Tabla XXXV. Atributos y métodos de la clase en el cliente ListadoMonitoreos.</i>	152
<i>Tabla XXXVI. Atributos y métodos de la clase en el cliente DetMonitoreoGen.</i>	152
<i>Tabla XXXVII. Atributos y métodos de la clase en el servidor DetallesMonitoreo.</i>	154
<i>Tabla XXXVIII. Atributos y métodos de la clase en el servidor ProcesarDetallesMonitoreo.</i>	
.....	154
<i>Tabla XXXIX. Atributos y métodos de la clase en el cliente MonitoreosAsoc.</i>	155
<i>Tabla XL. Atributos y métodos de la clase en el servidor RutaRetroalimentacion.</i>	155
<i>Tabla XLI. Atributos y métodos de la clase en el cliente RutaRetroalimentacion.</i>	155
<i>Tabla XLII. Atributos y métodos de la clase en el servidor MetricasAMonitorear.</i>	155
<i>Tabla XLIII. Atributos y métodos de la clase en el cliente MetricasAMonitorear.</i>	156
<i>Tabla XLIV. Atributos y métodos de la clase en el cliente AgregarNuevoElemento.</i>	156
<i>Tabla XLV. Atributos y métodos de la clase en el servidor ProcesarNvoElemento.</i>	156
<i>Tabla XLVI. Atributos y métodos de la clase en el cliente cajaElementos.</i>	157
<i>Tabla XLVII. Atributos y métodos de la clase en el cliente AdministrarProyecto.</i>	157
<i>Tabla XLVIII. Atributos y métodos de la clase en el servidor ProcesarAdministracion...</i>	158
<i>Tabla XLIX. Atributos y métodos de la clase en el cliente EquiposDeTrabajo.</i>	158
<i>Tabla L. Atributos y métodos de la clase en el servidor ProcesarEquiposDeTrabajo.</i>	159

LISTA DE TABLAS (CONTINUACIÓN)

	Página
<i>Tabla LI. Atributos y métodos de la clase en el cliente Requerimiento.</i>	<i>160</i>
<i>Tabla LII. Atributos y métodos de la clase en el cliente listadoRequerimientos.....</i>	<i>160</i>
<i>Tabla LIII. Atributos y métodos de la clase en el servidor listadoRequerimientos.....</i>	<i>161</i>
<i>Tabla LIV. Atributos y métodos de la clase en el servidor ProcesarRequerimiento.....</i>	<i>161</i>
<i>Tabla LV. Atributos y métodos de la clase en el cliente DetallesModulos.....</i>	<i>162</i>
<i>Tabla LVI. Atributos y métodos de la clase en el cliente listadoModulos.....</i>	<i>162</i>
<i>Tabla LVII. Atributos y métodos de la clase en el servidor ProcesarDatosModulos.</i>	<i>163</i>
<i>Tabla LVIII. Atributos y métodos de la clase en el servidor ListadoPruebas.....</i>	<i>163</i>
<i>Tabla LIX. Atributos y métodos de la clase en el cliente ListadoPruebas.</i>	<i>164</i>
<i>Tabla LX. Atributos y métodos de la clase en el cliente DetallesPruebas.</i>	<i>164</i>
<i>Tabla LXI. Atributos y métodos de la clase en el servidor ProcesarPruebas.....</i>	<i>164</i>
<i>Tabla LXII. Estructura de la tabla Proyectos.</i>	<i>165</i>
<i>Tabla LXIII. Estructura de la tabla tipoProy.</i>	<i>166</i>
<i>Tabla LXIV. Estructura de la tabla ciclos.</i>	<i>166</i>
<i>Tabla LXV. Estructura de la tabla Equipos.....</i>	<i>166</i>
<i>Tabla LXVI. Estructura de la tabla Usuarios_eq_rol.</i>	<i>166</i>
<i>Tabla LXVII. Estructura de la tabla Usuarios.</i>	<i>167</i>
<i>Tabla LXVIII. Estructura de la tabla usuariosNivelesLidederesProy.....</i>	<i>167</i>
<i>Tabla LXIX. Estructura de la tabla Requerimientos.</i>	<i>167</i>
<i>Tabla LXX. Estructura de la tabla relacinesReq.....</i>	<i>168</i>
<i>Tabla LXXI. Estructura de la tabla Modulos.</i>	<i>168</i>
<i>Tabla LXXII. Estructura de la tabla reqModulos.....</i>	<i>168</i>
<i>Tabla LXXIII. Estructura de la tabla modulosSubmodulos.....</i>	<i>169</i>
<i>Tabla LXXIV. Estructura de la tabla Modulos_mod.</i>	<i>169</i>
<i>Tabla LXXV. Estructura de la tabla procesosIS.....</i>	<i>169</i>
<i>Tabla LXXVI. Estructura de la tabla productosIS.....</i>	<i>169</i>
<i>Tabla LXXVII. Estructura de la tabla elementosIS.</i>	<i>169</i>
<i>Tabla LXXVIII. Estructura de la tabla Fases.....</i>	<i>170</i>

LISTA DE TABLAS (CONTINUACIÓN)

	Página
<i>Tabla LXXIX. Estructura de la tabla catalogoMetricasContextoIS.</i>	170
<i>Tabla LXXX. Estructura de la tabla catalogoMetricas.</i>	170
<i>Tabla LXXXI. Estructura de la tabla configuracionActualMetricas.</i>	170
<i>Tabla LXXXII. Estructura de la tabla relacionMetricas.</i>	171
<i>Tabla LXXXIII. Estructura de la tabla perfilesMetricas.</i>	171
<i>Tabla LXXXIV. Estructura de la tabla referenciasprocMedicion.</i>	171
<i>Tabla LXXXV. Estructura de la tabla recalendarizarAccionesProcMedicion.</i>	171
<i>Tabla LXXXVI. Estructura de la tabla Modelos.</i>	172
<i>Tabla LXXXVII. Estructura de la tabla parametrosModelos.</i>	172
<i>Tabla LXXXVIII. Estructura de la tabla valorParametrosModelos.</i>	172
<i>Tabla LXXXIX. Estructura de la tabla baseEstimaciones.</i>	172
<i>Tabla XC. Estructura de la tabla valoresEstimaciones.</i>	173
<i>Tabla XCI. Estructura de la tabla cfgMonitoreosActuales.</i>	173
<i>Tabla XCII. Estructura de la tabla tipoObjMonitoreo.</i>	173
<i>Tabla XCIII. Estructura de la tabla referenciaRacional.</i>	174
<i>Tabla XCIV. Estructura de la tabla racionalMonitoreos.</i>	174
<i>Tabla XCV. Estructura de la tabla tiposReferenciaRacional.</i>	174
<i>Tabla XCVI. Estructura de la tabla valoresAnomalos.</i>	175
<i>Tabla XCVII. Estructura de la tabla tiposValoresAnomalos.</i>	175
<i>Tabla XCVIII. Estructura de la tabla catalogoMonitoreos.</i>	175
<i>Tabla XCIX. Estructura de la tabla TipoEventos.</i>	175
<i>Tabla C. Estructura de la tabla Causas.</i>	176
<i>Tabla CI. Estructura de la tabla MasInformacion.</i>	176
<i>Tabla CII. Estructura de la tabla Rel_anomalia_causa.</i>	176
<i>Tabla CIII. Estructura de la tabla AccionRetroalimentacion.</i>	176
<i>Tabla CIV. Estructura de la tabla Metricas_a_monitorear.</i>	177
<i>Tabla CV. Datos del proyecto de prueba.</i>	178
<i>Tabla CVI. Ciclos de trabajo del proyecto de prueba.</i>	178

LISTA DE TABLAS (CONTINUACIÓN)

	Página
<i>Tabla CVII. Requerimientos del proyecto de prueba.</i>	<i>179</i>
<i>Tabla CVIII. Requerimientos del proyecto de prueba.</i>	<i>180</i>
<i>Tabla CIX. Módulos de diseño del proyecto de prueba.</i>	<i>182</i>
<i>Tabla CX. Módulos de implementación del proyecto de prueba.</i>	<i>182</i>
<i>Tabla CXI. Casos de prueba para el Requerimiento Funcional 1.</i>	<i>183</i>
<i>Tabla CXII. Métricas a monitorear durante el proyecto de prueba.</i>	<i>183</i>
<i>Tabla CXIII. Valores estimados para las métricas del proyecto de prueba.</i>	<i>184</i>
<i>Tabla CXIV. Valores actuales para las estimaciones de las métricas del proyecto de prueba.</i>	<i>184</i>
<i>Tabla CXV. Anomalías que se presentan para el primer ciclo de trabajo.</i>	<i>185</i>
<i>Tabla CXVI. Causas de las anomalías</i>	<i>185</i>
<i>Tabla CXVII. Acciones de retroalimentación para las interpretaciones halladas.</i>	<i>185</i>

Capítulo I.

Introducción

1.1. Introducción

Cada vez es mayor el número de proyectos de software de gran escala que se desarrollan comercialmente. Estos proyectos requieren la participación de diversos equipos de trabajo que ejecutan diferentes tareas del desarrollo. Es común que dichos equipos trabajan de forma paralela y geográficamente distribuidos. Cada equipo está encargado de desarrollar un subsistema particular o participar en alguna fase del proceso de desarrollo de software.

El enfoque de desarrollo distribuido posee ventajas tales como interactuar con personal altamente capacitado y aprovechar la diferencia de usos horarios. Este enfoque está teniendo un impacto profundo en la forma en que los productos se conciben, diseñan, construyen, prueban y se liberan al cliente. Sin embargo, también tienen desventajas como son la necesidad de metodologías ad hoc para su administración, herramientas para el intercambio y compartimiento de la información, y problemas de comunicación, entre otros.

Por otro lado, a medida que crecen los proyectos, crece su complejidad, es común que se salgan de control y estén fuera de las restricciones de tiempo, presupuesto, y calidad esperada.

Para tener un mayor control sobre el proyecto, se usa la administración de proyectos que se define como una actividad protectora dentro de la ingeniería de software que garantiza que las actividades se desarrollen en tiempo. Sus funciones principales son planificar, supervisar y controlar el trabajo de los ingenieros de software.

La administración juega un rol muy importante, muchas veces los fracasos de los proyectos se deben a una administración deficiente. Para que un producto de software sea exitoso, el administrador del proyecto debe integrar todos los aspectos asociados a este, tales como personal, clientes, etc. Las decisiones se hacen a diario, en como es que el producto técnico va a ser desarrollado y administrado, como es que los recursos serán asignados, así como los aspectos que deben ser tratados y la prioridad de estos. Mientras más efectivas sean las decisiones, es más exitoso el proyecto. A medida que los datos son más objetivos y se tiene disponible más información, serán mejor las decisiones.

Estas características se relacionan todas con la información y por lo tanto, están relacionadas a la medición. La medición auxilia al administrador del proyecto a hacer un mejor trabajo, ayudando a definir e implementar planes más realistas, a asignar recursos, monitorear acertadamente el progreso del proyecto y su relación con los planes. La

medición del software provee la información precisa para hacer la toma de decisiones del proyecto y tomar las acciones adecuadas.

Cuando los equipos de trabajo están distribuidos, cada uno de estos tienen un líder del equipo, estos equipos a su vez son guiados por un administrador quien tiene la tarea de conocer cual es la participación y efectividad con que cada equipo de trabajo esta contribuyendo en el desarrollo del proyecto, para lograr esto, es necesario monitorear el rendimiento de cada equipo a través de mediciones que reflejen cual es el desempeño de estos.

En la literatura se ha reportado una gran diversidad de guías, marcos de trabajo, metodologías y herramientas que demuestran el apoyo de la medición a la administración de proyectos. Los trabajos más completos brindan una gran diversidad de métricas que apoyan al administrador en todo un proceso de desarrollo y cubren algunas de las fases de lo que se conoce como un proceso de medición, el cual se define como un conjunto de actividades estructurado y repetible que permiten la medición en un proyecto dado, sus fases son: (1) formular que medir, (2) recolectar lo que se formulo, (3) monitorear las mediciones, (4) interpretación de los monitoreos, y (5) retroalimentación para conocer que cambios y toma decisiones es necesario de acuerdo a las mediciones obtenidas. Estas herramientas únicamente soportan la interacción entre el usuario y la herramienta, y es mínimo el soporte a la interacción entre usuarios, especialmente teniendo como base un proceso de medición. Este tipo de soporte es necesario para lograr la comunicación del equipo en apoyo al proceso de medición. Las aplicaciones colaborativas deben enfocarse en tres aspectos: (1) promover la participación entre el equipo del proyecto, (2) capturar, diseminar, y discutir la información y actividades del proyecto, (3) proporcionar un mecanismo de comunicación, estos puntos son importantes para herramientas que soporten a equipos de trabajo distribuidos.

En este trabajo se describe el análisis, diseño, e implementación de una herramienta que apoya el proceso de medición en el desarrollo de proyectos distribuidos con las siguientes características:

- Brindar mecanismos para que un administrador pueda administrar e interactuar con equipos de desarrollo.
- Hacer la distribución de actividades, subsistemas, y requerimientos para los equipos de trabajo.
- Proveer a los equipos de trabajo y al administrador de proyectos una jerarquía de métricas asociadas al proyecto de desarrollo y a los productos que genera cada equipo de trabajo.
- Cubrir las fases de formulación, recolección, monitoreo, interpretación y retroalimentación del proceso de medición y permitir que los equipos y a su vez los ingenieros participen en dicho proceso de forma colaborativa y con responsabilidades compartidas.

1.2. Planteamiento del problema

El administrador requiere conocer el status en el que se encuentra cada equipo de trabajo. Debe contar con mecanismos para monitorear el trabajo de los equipos y que a su vez estos equipos tengan mecanismos para reportar el trabajo. Las métricas de software sirven para que un administrador tenga una visión del desempeño de los equipos de trabajo.

Desafortunadamente no hay una única métrica que se pueda aplicar a todo un desarrollo de software. Se tienen diversos tipos de métricas, según se apliquen al proyecto, procesos, productos y recursos del desarrollo.

En cuanto a métricas de apoyo a la administración, es lamentable que no se tenga un conjunto de métricas aceptado universalmente. Sin embargo, en la academia y en la industria, el administrador de proyectos se apoya de conjuntos de métricas que se han definido por de facto.

La pregunta que surge es: ¿qué elementos se deben considerar para brindarle al administrador del proyecto un conjunto de métricas que le brinden información de cual es el status que mantienen equipos de desarrollo distribuidos geográficamente? Para responder a esta pregunta es necesario ver a la administración y la medición desde un nivel en el que su funcionamiento no dependa de una metodología de desarrollo.

Una limitación de la mayoría de las herramientas es que solo proveen las métricas sin el contexto de otras actividades de la administración del proyecto, por ejemplo, no están asociadas a ciclos de desarrollo, o a los productos de trabajo que se liberan, o a los equipos de trabajo que participan en el desarrollo. Otra carencia es que muy pocas herramientas cubren las fases del proceso de medición y tampoco abarcan la participación y colaboración de los equipos de desarrollo en dicho proceso.

1.3. Objetivo del trabajo de tesis

El objetivo principal de este trabajo es crear una guía que apoye a un administrador de proyectos en conocer el status de equipos de trabajo distribuidos según las métricas de software que estos registran dependiendo de su participación que tienen en el proyecto.

Los ingenieros de software deben participar de forma colaborativa en el registro y monitoreo de las métricas, para lograr así mayor efectividad en el uso de estas mediciones.

Para desarrollar esta herramienta se realizó un análisis sobre los diferentes marcos de trabajo que brindan apoyo a un administrador a través de mediciones y de la administración de equipos de trabajo, de estas mejores prácticas resultaron los requerimientos necesarios para el diseño e implementación de una herramienta de apoyo a la administración de equipos distribuidos a través de las métricas de software que estos generan.

I.4. Organización de la tesis

En el capítulo II se describen los fundamentos asociados a la medición, medida e indicadores, los tipos de métricas, el proceso de medición, el plan de medición. Las métricas de la administración de proyectos, los marcos de trabajo y las herramientas existentes.

En el capítulo III se describe el proceso de medición para equipos de trabajo distribuidos, se definen elementos tales como: principios, componentes de información, implementación del proceso de medición y métricas.

El capítulo IV se refiere al análisis y diseño del Sistema de Apoyo al pRoceso de mEdición (SARE) basado en equipos de trabajos.

En el capítulo V se describe la implementación tecnológica de SARE y también se presentan la descripción del sistema a través de pantallas.

El capítulo VI presenta las pruebas que sirvieron para validar que los requerimientos para SARE se hayan cumplido.

Por último en el capítulo VII se presentan las conclusiones, aportaciones y trabajo a futuro.

Capítulo II.

Fundamentos

II.1. *Introducción*

Las métricas de software tienen una diversidad de aplicaciones que van desde métricas particulares para medir el desempeño personal hasta indicadores que evalúan el nivel organizacional de una empresa. Esta diversidad comprende una amplia referencia literaria y una diversidad de términos. Sin embargo, en la literatura de métricas hay conceptos que es necesario definir sin importar el dominio de aplicación.

En este capítulo se describen estos conceptos generales así como los que permitirán entender mejor el trabajo de tesis, es decir, conceptos que hablan de la relación entre las métricas con la administración de proyectos y equipos de trabajo distribuidos. También se habla del trabajo previo y el estado del arte actual así como las herramientas de brindan apoyo a la medición.

II.2. *Métrica, medición e indicadores*

En la literatura de métricas de software es común encontrar los términos métrica, medición, e indicador.

Una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto de software [Pressman, 2002]. Por ejemplo, una medida son las líneas de código. Aunque no se tiene un estándar universal para las líneas de código, si puede tenerse un estándar local de cómo se miden las líneas de código dentro de una organización.

El Instituto de Ingenieros Electrónicos (IEEE por sus siglas en inglés) en su glosario estándar de términos de la Ingeniería de Software define una métrica como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [Ejiogo y Len, 1991]. Según Pressman una métrica es un valor calculado o compuesto por dos o más mediciones [Pressman, 2002]. Por ejemplo, una métrica son el número de defectos por miles de líneas de código.

Un indicador es una combinación de métricas que proporcionan una visión profunda del proceso, proyecto o producto de software [Pressman, 2002]. Un indicador generalmente compara una métrica con una línea base o un valor esperado. Un ejemplo de indicador esta dado por un incremento substancial del número de defectos encontrados en la liberación más reciente del código. La Figura 1 ilustra la relación entre los conceptos medición, métrica e indicador.

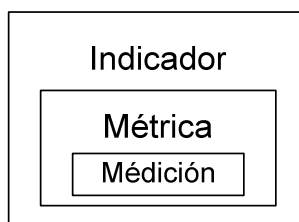


Figura 1. Relación entre los términos medición, métrica e indicador

El término medición es el acto o proceso de determinar una medida. Fenton define formalmente este término como “el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas” [Fenton y Pfleeger, 1997].

Sin embargo, Fenton habla del concepto de métricas de software de forma mas generalizada. “Métricas de software es un término colectivo que se usa para describir la amplia variedad de actividades de la medición en la ingeniería de software” [Fenton y Pfleeger, 1997].

Para no entrar en confusión entre los términos medición, métrica e indicador que intuitivamente hablan de lo mismo, durante el trabajo de investigación cuando se hable de métricas de software será en el sentido que habla [Fenton y Pfleeger, 1997] y de forma indistinta se utilizarán los conceptos de métrica e indicador.

II.3. Tipos de métricas

Las métricas pueden utilizarse para una diversidad de propósitos. Por ejemplo, determinar los niveles de desempeño actuales o la predicción de dichos niveles. Dependiendo del propósito, se tienen diferentes tipos de mediciones. En la literatura, las métricas se han clasificado en los siguientes tipos:

- Métricas de productos y del proceso.
- Métricas objetivas y subjetivas.
- Métricas directas e indirectas.
- Métricas explícitas y derivadas.
- Métricas absolutas y relativas.
- Métricas dinámicas y estáticas.

Sin embargo, no todas las clasificaciones anteriores se utilizan frecuentemente. A continuación se describen los tipos que se usan comúnmente.

II.3.1. Métricas del proceso y del producto

Generalmente dentro de un desarrollo de software, las métricas de software, se pueden clasificar en métricas del proceso, del producto, de recursos:

- Las métricas del proceso cuantifican los atributos del proceso de desarrollo y del ambiente de desarrollo. Un ejemplo es el número de defectos que se cometen durante el proceso de desarrollo.
- Las métricas del producto mide los atributos del producto de software. Se enfocan en los documentos de requerimientos, diseño, o código fuente. Ejemplos de estas métricas son el número de requerimientos y la complejidad del código.
- Las métricas de los recursos miden las entradas requeridas para una actividad de la ingeniería de software tal como lo es el hardware, software, documentación, conocimiento, y recursos humanos.

Pressman hace otro nivel más de clasificación al incorporar las métricas del proyecto [Pressman, 2002]. Estas métricas cubren datos asociados a todo el desarrollo del proyecto, tal como lo son la duración y esfuerzo (personas-mes) requerido por el proyecto.

Las entidades del proyecto, procesos, productos y recursos (PPPR) corresponde a los elementos ortogonales de la ingeniería de software, al complementarse integran un proceso de desarrollo.

II.3.2. Métricas objetivas y subjetivas

Las métricas objetivas son mediciones absolutas que se obtienen del proceso o producto, y contabilizan atributos o características de una forma objetiva [Humphrey, 1995]. Por ejemplo líneas de código y número de defectos descubiertos. Estas métricas tienen un punto de inicio fundamental, un cero natural. Las métricas objetivas deberían brindar siempre resultados idénticos para una medición que cuando se realiza por dos o más observadores calificados con las mismas cualidades.

Las métricas subjetivas son mediciones de un proceso o un producto que envuelve un juicio humano y subjetivo. Un ejemplo de esta métrica es la complejidad esperada del código.

II.3.3. Métricas directas e indirectas

Una métrica directa es una medición de un proceso o características del producto que no dependen de la medición de alguna otra característica. Ejemplos son el número de defectos en un producto o bien el número de días que dura un proceso.

Una métrica indirecta, es una medición que involucra una o más características. Por ejemplo, la densidad de defectos se calcula como el número de defectos sobre el tamaño del producto.

II.3.4. Ciclo de vida

Las métricas pueden también clasificarse según el ciclo de desarrollo. Pueden clasificarse como métricas de planeación, requerimientos, diseño, codificación y pruebas.

II.3.5. Herramienta

Una herramienta para la evaluación y medición se le conoce como CAME (por sus siglas en inglés, Computer Assisted Software Measurement and Evaluation) puede brindar apoyo automático, manual o semimanual, según el apoyo que brinda es el tipo de métrica, es decir, automático, manual o semimanual, respectivamente.

En el trabajo de investigación se incorporan las métricas de la administración de proyectos en un ambiente de desarrollo distribuido de software. La visión que tiene un administrador es sobre todos los elementos del desarrollo del software, por lo tanto, la clasificación que resulta más conveniente es por proyecto, procesos, productos y recursos. De hecho, esta clasificación cubre la taxonomía por los ciclos de vida.

Independientemente del tipo, para hacer un uso más efectivo de las métricas, no es suficiente con creer que las actividades de medición se terminan con la recolección de los datos. Es necesario representar (visualizar) e interpretar los resultados de las mediciones y ver que cambios implican dentro de un proceso de desarrollo de software.

Las actividades anteriores, se enmarcan en lo que se conoce como un proceso de medición.

II.4. *Proceso de medición (PM)*

Es inútil hablar del valor agregado de la medición cuando esta no está ligado al proceso de desarrollo de software. La razón es simple, las métricas obligan a conocer como es que se está llevando el proceso de desarrollo de software. El proceso de medición es la capa que interactúa con el contexto de desarrollo en el se sitúan las mediciones.

Cuando las métricas tienen alcance a un contexto de aplicación, es decir, a los elementos del proceso de desarrollo, es posible responder a las preguntas ¿qué?, ¿sobre qué?, ¿quién?, ¿qué equipo?, entre otras. Los pasos del PM son: formulación, recolección, análisis, interpretación y retroalimentación.

II.4.1. Formulación

En esta fase, se planean las métricas apropiadas para la representación del software y para los objetivos que se desean alcanzar, estas métricas deben adaptarse a los productos y procesos específicos.

El problema que se tiene en esta fase es responder a la pregunta ¿qué entidades del proceso de desarrollo se deben medir de modo que se cubra con los objetivos planteados?

El paradigma GQM (Goal-Question-Metric) ayuda a decidir que mediciones tomar y cómo utilizarlas, para el uso de este paradigma es necesario identificar los siguientes elementos:

1. Metas. Una meta se establece al contestar la pregunta ¿qué es lo que la organización esta tratando de alcanzar? Una mejora del proceso se logra cuando son satisfechas dichas metas.
2. Preguntas. Aquí se establecen las preguntas (áreas de incertidumbre) que requieren ser contestadas para que se cumpla la meta.
3. Métricas. Son las mediciones que deben recolectarse para brindar respuesta a una pregunta y confirmar si las mejoras del proceso ayudan a cumplir la meta deseada.

El enfoque GQM separa las cuestiones organizacionales, es decir las metas, de las cuestiones específicas del proceso representadas por las preguntas.

Lo que no es evidente del paradigma GQM es el modelo que se necesita para combinar las mediciones de modo que las preguntas tengan respuesta. Por ejemplo, la productividad del codificador puede ser medida en términos de esfuerzo por línea de código, pero dicha relación no es explícita en el árbol GQM. Por lo tanto, el enfoque GQM debe ser suplantado por uno o más modelos que expresen las relaciones entre las métricas, o bien las relaciones de las métricas respecto a las entidades del desarrollo.

II.4.2. Recolección

Es el mecanismo empleado para obtener los datos necesarios que cubran con las métricas establecidas en la fase de formulación. Estos son datos crudos que se almacenan en una base de datos de mediciones. Si se recolectan datos subjetivos, los datos deberían ser validados antes de ser almacenados.

Si los ingenieros recolectan los datos manualmente, entonces los formularios se completan durante la recolección de los datos. Idealmente, un manual del proceso debe describir cuando es que estos cuestionarios deben ser completados, también un plan de recolección de datos debe indicar cuando es que determinadas métricas requieren ser recolectadas.

De ser posible la recolección debe automatizarse. Dada la diversidad de elementos del desarrollo, plataformas y productos de apoyo a la Ingeniería de Software (tipo CASE, por sus siglas en inglés), desafortunadamente esta automatización no puede ser cubierta con una sola herramienta.

II.4.3. Análisis

En esta fase se procesan los datos crudos que se capturaron en la etapa de recolección. Con estos datos se forman las métricas que se requieren (según la fase de formulación) y se elaboran gráficas y reportes que se utilizarán en la etapa de interpretación. Estas gráficas y reportes deben enfocarse a las metas planteadas.

Los reportes generalmente presentan información tabulada, en cuanto a las gráficas, estas son de diversos tipos y deben mostrar claramente los valores que están fuera de los rangos establecidos.

Los reportes y las graficas se complementan, es común que un reporte incluya diversas graficas.

El problema de esta fase es que no se tiene estandarizado la cantidad y tipo de reportes y gráficas que se deben producir; sin embargo ambos están en función de los objetivos planeados. Más adelante se verá que hay muchos reportes y graficas que se han utilizado por de facto, según el dominio de aplicación de las métricas. Un ejemplo es el reporte semanal del status del proyecto.

II.4.4. Interpretación

La interpretación de los datos que se miden debe realizarse en el contexto de las metas y los resultados que se espera alcanzar. Se deben establecer directivas de interpretación sobre las gráficas y los reportes. Estas directivas se basan en: (1) recomendaciones de la literatura, (2) memoria organizacional y (3) experiencia de los ingenieros.

- Las recomendaciones de la literatura establecen relaciones entre las métricas, estas relaciones guían para conocer las causas de los problemas. Un ejemplo es el trabajo de Kitchenham, donde provee una serie de causas tanto buenas como malas de lo que posiblemente ha generado problemas o beneficios al proyecto [Kitchenham y Pickard, 1987].
- La memoria organizacional sirve para llevar un registro de las interpretaciones y los problemas que se han presentado en algún proyecto y que posiblemente se estén repitiendo en uno nuevo.
- La experiencia de los ingenieros provee información desde el punto de vista del desarrollador de cuales fueron las causas de los problemas, estas pueden efectivamente ser las causas reales, o bien, suposiciones de dichas causas.

En todas las directivas de las métricas, a medida que se establecen relaciones entre las entidades, es posible establecer más causas de los problemas.

Se organiza una junta donde por medio de los reportes y las graficas se comunican los datos de las métricas, después se descubren y discuten las causas de los problemas.

A medida que se tiene más información, es necesario que se presenten más juntas, de modo que se analice toda la información. Al final de la fase de interpretación, se tiene conocimiento de las áreas más problemáticas del proyecto. Estos problemas se discuten en la fase de retroalimentación.

II.4.5. Retroalimentación

En esta fase se determinan según la información de las fases anteriores, cuales serán las recomendaciones para la mejora del proceso de desarrollo.

Se registran los cambios al proceso y es posible reestablecer nuevas metas y objetivos que se pretendan alcanzar. Se deben seguir los siguientes principios:

- Definir las formas en que los resultados de las métricas dirigen el cambio en el proceso.
- Definir los datos que formaran la línea base y determinar conjuntos de procedimientos para reestablecerla. Se debe asegurar que todos los niveles del equipo provean sugerencias a los cambios en el proceso.

Para cambiar con éxito el proceso, significa hacer que todos conozcan los cambios que se requiere realizar, esto puede ser a través de juntas, memorandums, o correos.

Las fases de análisis, interpretación y retroalimentación mantienen una relación muy estrecha. Las gráficas y reportes que se interpretan se relacionan a un problema particular, estos problemas tienen asociadas circunstancias particulares (causas) y estas a su vez originan cambios que se esperan en el proceso de desarrollo.

En resumen, después del proceso de medición se tienen los resultados de las métricas, junto a las experiencias del proyecto que debieron recolectarse y administrarse sistemáticamente para uso futuro. Los resultados deberían incorporarse en la administración de la experiencia de la organización (memoria organizacional). Este contexto común es la base para el aprendizaje y mejora continua de los proyectos, formando así una referencia de calidad que sirve para predecir esta según las características iniciales del proyecto.

II.5. Plan de medición

Un plan de métricas debe describir el quien, que, donde, cuando, como y porque de las métricas. Con respuestas a todas estas preguntas, quien lea el plan conoce exactamente porque es que las métricas están siendo recolectadas, como se están usando, y como las métricas se incorporan al panorama de desarrollo.

El plan usualmente inicia con el porque. Es en esta sección introductoria del plan cuando se exponen las metas u objetivos del proyecto, se describen que respuestas necesitan ser respondidas por los miembros del proyecto y el administrador del proyecto. Por ejemplo, si la fiabilidad es la principal preocupación de los desarrolladores, entonces el plan discute como es que la fiabilidad va a ser definida y que requerimientos deben tener los reportes del sistema; en secciones posteriores del plan se discute como la fiabilidad va a ser medida y como es que será monitoreada.

Después, el plan considera que es lo que va a ser medido. En muchos casos, las medidas son agrupadas o relacionadas de alguna forma. Por ejemplo, la productividad podría ser medida en términos de dos componentes: tamaño y esfuerzo. Por lo tanto el plan va a explicar como es que se definen el tamaño y el esfuerzo, así como es que estas métricas se combinan para calcular la productividad.

También, el plan debe exponer el donde y cuando se llevará a cabo el proceso de medición. Algunas mediciones se pueden realizar una sola vez, mientras que otras se hacen repetidamente y se le monitorea a través del tiempo. El tiempo y la frecuencia de recolección están relacionados a las metas y necesidades del proyecto, y estas relaciones deben estar explícitas en el plan.

Es importante además de discutir que medidas se necesitan, también definir que herramientas van a ser usadas para capturar y almacenar datos, así como quien es el responsable de dichas actividades. Frecuentemente, los planes de métricas ignoran estos aspectos cruciales, y los miembros del proyecto asumen que alguien más esta cuidando de las métricas; el resultado es que todo mundo esta de acuerdo en que las métricas son importantes, pero nadie esta asignado a la tarea de la recolección de los datos [Fenton y Pfleeger, 1997]. Si la responsabilidad para el análisis no la adopta alguien, entonces solo se forma una pila de datos que nunca es usada en la toma de decisiones. El plan debe establecer claramente que tipos de análisis son los que se realizarán con los datos, quién hará el análisis, y como se presentan los resultados al personal encargado de la toma de decisiones, y como es que estos resultados apoyaran a las decisiones.

Por lo tanto, el plan de métricas describe un panorama completo del proceso de medición, desde la definición inicial de las necesidades, pasando por el análisis y hasta finalmente la aplicación de los resultados.

II.6. Métricas de la administración de proyectos

Las métricas son herramientas poderosas, pero no son populares con administradores que están atrapados en proyectos y circunstancias difíciles. No obstante se tiene un conjunto de métricas básicas que comúnmente se aplican, tal como la duración y el esfuerzo, [Putnam y Myers, 1992]. Los programas de métricas usualmente fallan cuando intentan medir demasiado. Cuando es necesaria una excesiva recolección de datos, es común falsificar los números para ahorrar tiempo.

Según [Putnam y Myers, 1992] lo que se mide es:

1. El tamaño del producto.
2. El esfuerzo requerido para hacer el producto.
3. La calidad del producto.
4. La duración.

El producto difiere según las diferentes fases del proyecto. Mientras se recolectan los requerimientos, el producto es la documentación de los requerimientos. El producto en la fase de diseño es el documento de diseño. El código fuente es el producto en la fase de implementación. Los reportes de pruebas son los productos de las fases de pruebas.

II.6.1. Tamaño

La medida del tamaño varía de acuerdo al producto. El tamaño de los requerimientos es la contabilización de los requerimientos individuales. Para el diseño la medida del tamaño es la contabilización de los elementos de diseño (sistemas, módulos). En la etapa de implementación el tamaño son las líneas de código, las subrutinas, o clases que integran el sistema (para sistemas orientados a objetos). Finalmente la medida del tamaño en las pruebas son el número de casos de pruebas.

II.6.2. Esfuerzo

El esfuerzo asociado con un producto se mide en términos de personas-unidades de tiempo (horas, mes). El plan del proyecto estima las personas-horas necesarias para crear una unidad del producto.

II.6.3. Calidad

La calidad del producto se basa en los errores que se encuentran y se corrigen. Los ingenieros hacen y descubren errores en todas las fases de un proyecto, lo hacen a través de las inspecciones y no solo en las pruebas.

II.6.4. Duración

La duración es el periodo entre que una tarea inicia y termina.

II.7. *Marcos de trabajo existente*

La medición práctica de sistemas y software (PSM por sus siglas en inglés) y el proceso para equipos de software (TSP por sus siglas en inglés) son las dos propuestas que fuertemente llevan las métricas de software con el fin de administrar proyectos. PSM esta fuertemente ligada solo a la medición sin proveer un contexto del desarrollo de software, y en contraparte, TSP si provee un contexto de desarrollo de software enfocado a equipos de trabajo y lo liga a la medición solo que con un menor nivel de detalle que lo hace PSM.

II.7.1. PSM

PSM fue patrocinada por el Departamento de Defensa de la Armada de Estados Unidos para cumplir con los retos técnicos y administrativos que implican el desarrollo del software actual. Es un proceso de medición guiado por las necesidades de información que guía a cumplir las metas de negocios de la organización. PSM se enfoca a cubrir necesidades requeridas por un administrador de proyectos. La guía en PSM representa las

mejores prácticas usadas por los profesionales de la medición, las características principales son:

- Proveer a los administradores de proyectos con información objetiva y necesaria para cumplir satisfactoriamente objetivos técnicos, costo y calendario.
- Se basa en las mejores prácticas del Departamento de Defensa, gobierno y programas de la industria.
- Es un proceso flexible.
- Define un enfoque de análisis basado en la información.
- Provee una base para la administración a nivel empresarial.
- Es compatible con el estándar ISO/IEC 15939 para el proceso de medición del software.

PSM es el estándar de facto para cubrir las necesidades de información de un administrador de proyectos, prueba de esto es la implementación que se hace de PSM dentro del Proceso Unificado de Rational (RUP por sus siglas en inglés).

II.7.2. TSP

TSPi provee principios de la Ingeniería de Software (IS) para el trabajo en grupos, define roles explícitos y las responsabilidades que estos tienen.

II.7.2.1. Principios de TSPi

- TSPi obliga a un aprendizaje rápido que permite obtener una rápida retroalimentación. Los scripts TSPi proveen un marco de trabajo definido, medible, y repetible para los equipos.
- La productividad del equipo requiere “metas específicas”, un “ambiente de trabajo de apoyo”, así como un buen entrenador y un líder.
- Para la solución de problemas, se requieren bases sólidas, no es conveniente perder el tiempo definiendo prácticas, métodos, y roles.
- Descripción detallada del proyecto y los procesos que se usan, apoyo a planeación, seguimiento, y administración de la configuración. Procedimientos para la definición de estándares para requerimientos, diseño, implementación y pruebas. Así como métodos y procedimientos para inspecciones y revisiones.

II.7.2.2. Principios del diseño TSPi

1. Construir un marco de trabajo que se fundamente en el proceso de software personal conocido como PSP (PSP básicamente son principios de IS para que un programador controle su desempeño personal), TSP no es abrumante, ya que las tareas son específicas de cada rol. Estas son tareas de planeación, seguimiento de la calidad, apoyo y liderazgo.

2. Desarrollar productos en diversos ciclos. Los ciclos permiten identificar problemas de diseño, desempeño, y usabilidad de cada ciclo.
3. Establecer medidas estándar de calidad y desempeño. Las medidas para evaluar la calidad de un trabajo dado las brinda PSP, TSP dice como interpretar las medidas de PSP y como aplicarlos al proyecto.
4. Proveer medidas precisas para equipos. Con esto se conoce el desempeño personal de los miembros del equipo, esa es la naturaleza de un equipo, conocer los métodos y esperar comentarios razonables.
5. Usar roles y evaluaciones de los equipos. Opcionalmente se pueden hacer evaluaciones de los compañeros, se deben hacer evaluaciones de cómo es que el rol repercute en el proceso y no de la gente.

Requerir una disciplina del proceso. Los beneficios de TSPi solo se obtienen cuando se completan los formularios, se analizan y usan sus datos.

TSP se basa en un desarrollo por ciclos o evolutivo, consiste en dividir el proceso de desarrollo en iteraciones y asigna actividades a cada ciclo. Un contexto de este tipo promueve el uso de las métricas de software, por ejemplo, es posible conocer el número de requerimientos que serán desarrollados en un ciclo particular. También cada vez que inicia un ciclo se pueden programar reestimaciones o recolecciones de datos, cuando monitorear, o bien, cuando planear nuevas métricas.

Denger habla de las ventajas del enfoque evolutivo, como un enfoque que se basa en la premisa de que los clientes y la administración juzgan los proyectos y productos según evidencia de los resultados que se entregan [Denger, 1990]. Por lo tanto, la idea básica es planear liberaciones frecuentes de un sistema. De esta forma los requerimientos del proyecto son administrados fácilmente.

Este enfoque permite a los equipos:

- Centrarse en metas claras y a corto plazo con un alto grado de confianza en que estas sean satisfechas.
- Asegurar que se obtuvo una retroalimentación al final de cada entrega.
- Mantenerse en control con la producción.
- Facilidad de estimación al contar con ciclos de producción cortos.

II.8. Herramientas

La clasificación de las herramientas de medición puede verse desde múltiples puntos de vista. No es útil basarse en una sola clasificación para discernir que herramienta utilizar. La selección debe estar en función según se complementen diferentes criterios, esto es, diferentes taxonomías.

En el trabajo de investigación interesa conocer las herramientas CAME desde los siguientes puntos de vista:

- Fases del proceso de medición
- Productos del proceso de desarrollo
- Administración de proyectos
- Utilidades de apoyo

Dumke hizo un estudio extenso de las herramientas de apoyo a la medición, este estudio clasifica las herramientas según el tipo de métrica y sirve de base para la siguiente descripción de las herramientas [Dumke et al, 1997].

II.8.1. Fases del proceso de medición

En general, cualquier herramienta de métricas puede incluirse en esta clasificación; sin embargo, hay herramientas que evidentemente le brindan mas apoyo a alguna fase en particular, ¿cuáles son los mecanismos de apoyo para cada fase del PM?, a continuación se describen estos mecanismos y herramientas.

II.8.1.1. Formulación

En esta fase, las herramientas pueden ser de dos tipos. El primero son aquellas herramientas que evalúan la capacidad de desarrollo que tiene la organización. Soft-Org evalúa dicha capacidad y AMI construye árboles de metas de lo que se recomienda medir. El segundo son herramientas que permiten definir métricas. Athena define métricas, a través de estructuras léxicas y sintácticas, también Cosmos define un lenguaje de especificación de mediciones además de tener métricas predefinidas.

II.8.1.2. Recolección

Las herramientas de recolección de datos, básicamente capturan mediciones, estas pueden ser capturadas manual o automáticamente.

En la captura manual es necesario que el usuario registre los datos de la medición. Un ejemplo es cuando se reportan defectos, es común que las características de estos se registren manualmente tal como lo hace Bugzilla [Bugzilla]. En Prometrix el usuario es responsable de modelar un grafo del flujo de control del programa y este sirve para obtener métricas automáticamente.

En la captura automática las herramientas que utilizan este mecanismo son diversas, en especial las que procesan código fuente, un ejemplo es STW-Metric, JMetric captura informes de programas orientados a objetos. De hecho cualquier compilador puede considerarse en esta categoría. STW-Metric hace pruebas del software a diferentes niveles.

No hay herramientas que solo recolecten o reciban como entradas datos y no regresen nada de resultados, generalmente todas las herramientas de recolección, realizan análisis y reportan datos a través de gráficas y reportes (tablas).

II.8.1.3. Análisis

Estas herramientas ofrecen diferentes opciones para preparar gráficos y reportes. Muchas de estas herramientas también abarcan la fase de recolección.

JMetric presenta una gráfica de las mediciones. Prometrix presenta los resultados en tablas o en simples diagramas y STW-Metric reporta los resultados de las pruebas a través de diagramas de Kiviat.

II.8.1.4. Interpretación

Para la interpretación de los resultados es importante que se hayan administrado los resultados de la medición del proyecto, puede verse como una precondition que se tenga alcance a una base de datos de mediciones.

Existen múltiples herramientas que apoyan a la interpretación de los datos, generalmente esta interpretación se da en términos de la calidad que se ha alcanzado en el proyecto. Las herramientas Datrix y LDRA Testbed comúnmente hablan de calidad en términos del código y facilidad de pruebas, generalmente los valores que se obtienen se compara con los recomendados.

En general las herramientas de apoyo a la interpretación cumplen con las siguientes características:

- Se auxilian de una base de datos empíricas (memoria organizacional, experiencias propias).
- Recomendaciones citadas en la literatura.
- Límites establecidos según el usuario (valores máximos y mínimos).

En general, no hay herramientas que cubran el proceso de medición de inicio a fin, lo que si hay son herramientas que cubren varias fases. Por ejemplo, es difícil hallar herramientas de recolección de datos que no reporten los resultados de las métricas, es decir, que estén ligadas a la fase de análisis.

Las herramientas principalmente se dividen en herramientas de formulación, recolección y análisis e interpretación.

La etapa de retroalimentación no cuenta con herramientas que directamente la apoyen. Esto se debe a que esta etapa tiene que ver más con el proceso de desarrollo que con las mediciones en sí, aunado a la particularidad de cada proceso de desarrollo y de que la retroalimentación es más bien una inspección de los resultados de las métricas, todo esto hace más difícil encontrar herramientas que cubran esta fase.

En las fases de interpretación y retroalimentación es necesaria la discusión de resultados, a medida que se discuten se obtiene un mayor beneficio de estas fases. No se encontraron herramientas que brinden particularmente apoyo a las discusiones de estas fases. Una

herramienta de este tipo resultaría útil que se base en un ambiente de equipos de trabajo, tal como lo es TSPi.

A medida que las herramientas tengan un conjunto de características muy específicas es común que estas se enfoquen solo a una fase en particular, tal como sucede en las herramientas de definición de métricas, donde inclusive se necesita pasar la información a un formato intermedio y este es el que es manipulable.

Para resolver la carencia de herramientas que cubran el proceso de medición de inicio a fin, actualmente lo que se ofrece es una serie de herramientas que interactúan entre sí y que en conjunto cubren las fases de formulación, recolección y análisis e interpretación. Otra opción que se provee es ofrecer herramientas que se enfocan a una fase y que mantienen interfaces de comunicación (plug-ins) a herramientas fabricadas por terceros.

Dada la diversidad de herramientas que se tienen para cada fase del proceso, otro camino para cubrir el proceso de medición es implementar administradores de cada uno de los productos que se obtienen en cada fase del proceso, es decir, hablar de una administración de la configuración sobre el proceso de medición

II.8.2. Productos del proceso de desarrollo

La clasificación de las herramientas que cubren productos particulares muestra claramente como la diversidad de factores que ocurren en un ambiente de desarrollo hacen complejo contar con una herramienta genérica que abarque todos los ambientes y productos del desarrollo de software.

Por otro lado, los productos durante el desarrollo del producto demuestran el avance del proyecto. A medida que se completan más productos se tiene más avance en el proyecto.

Se pueden obtener métricas de estos productos a través de dos tipos de herramientas:

- Herramientas que administran los productos y que implícitamente proveen métricas tal como el número de requerimientos.
- Herramientas que evalúan determinadas métricas sobre los productos

A continuación se muestran ejemplos de productos, según la fase de desarrollo.

II.8.2.1. Análisis

Generalmente estas herramientas no dependen de una metodología de desarrollo en particular, solo consisten en administrar u obtener una estimación de la calidad de los requerimientos.

II.8.2.1.1. Administración (rastreo) de requerimientos

Las herramientas de requerimientos enlistan los requerimientos basados en documentos de texto o en herramientas CASE. Se pueden usar para contar los requerimientos y monitorear la evolución de los requerimientos. Un ejemplo es TWK/RTRACE de Cadre Technologies, RequisitePro de Rational Rose.

II.8.2.1.2. Calidad de requerimientos

La calidad de los requerimientos se conoce a través del número y tipo de defectos contenida en un documento de requerimientos. La herramienta ARM de la NASA [Wilson et al, 1997], analiza requerimientos a través de un análisis sintáctico (previo lexicón definido), en base a ciertas palabras claves se conoce aspectos de la calidad de los requerimientos.

II.8.2.2. Diseño

Hay diversos tipos de herramientas según la metodología de desarrollo, un ejemplo es la herramienta Rose que devuelve métricas referentes a diseños orientados a objetos que se hacen usando la herramienta.

Otro ejemplo es Battlemap que es una herramienta que mide el flujo de control de los módulos y la jerarquía de llamadas entre módulos. Demeter también apoya las características estructurales de los módulos.

II.8.2.3. Implementación

Al igual que el diseño, dependen del tipo de programación: estructurada u orientada a objetos, principalmente son herramientas de análisis de código. Estas herramientas están disponibles para contar las líneas de código, líneas de documentación, y analizar la complejidad del código.

Los programas para contabilizar código son programas que aceptan un archivo de entrada y contabilizan un número de líneas no blancas y no comentadas, así como el número de líneas comentadas. La definición de una línea de código varía entre las organizaciones, básicamente la regla es ser consistentes.

Las herramientas de análisis de complejidad revisan el código fuente y estiman la complejidad del software.

Ejemplos son la herramienta CHECKPOINT de Software Productivity Research y las herramientas de McCabe de McCabe y Asociados.

También están las herramientas para análisis de código orientado a objetos. Un ejemplo es JMetric la cual además presenta gráficamente las mediciones. OOMetric además de analizar código compara los resultados obtenidos con los recomendados (empíricos) o los valores límite según el usuario.

II.8.2.4. Pruebas

Hay herramientas muy específicas que inclusive están integradas a ambientes de desarrollo y que generan casos de prueba particulares, tal es el caso de JMetric, también hay herramientas de administración de casos de prueba y por último aquellas herramientas que ven la calidad del software a través de las pruebas.

II.8.2.4.1. Administración de pruebas

Las herramientas de administración de pruebas apoyan la enumeración, status y ejecución de pruebas y pruebas de regresión. Estas herramientas identifican cuales son las ramas o enunciados del programa que son ejecutados por un conjunto de casos de prueba. Se pueden usar las métricas del programa para recolectar datos de las pruebas. Ejemplos incluyen TestManager, TestPro, y herramientas de McCabe de McCabe y Asociados.

II.8.2.4.2. Estimación de calidad

LDRA Testbed ofrece diferentes reportes y criterios de estimación de calidad, según criterios sobre el código y si están conformes a los estándares de programación, esto se refleja en las consecuencias que involucra la complejidad de las modificaciones.

II.8.3. Administración de proyectos

Estas herramientas asisten al administrador de proyectos a llevar control sobre diferentes actividades, todas estas herramientas se enfocan básicamente a responder a la pregunta ¿cuál es el status del proyecto en términos de tiempo, costos y personal?

II.8.3.1. Herramientas de estimación de costos

Las herramientas de estimación de costos estiman el tamaño del software en SLOC o los puntos de función como la entrada según el número de parámetros que influyen la productividad. Se utilizan modelos de estimación de costos (tal como COCOMO 2) para estimar el esfuerzo, costo, tasa de trabajo, perfiles del personal y agenda para cada actividad del software. Ejemplos que se incluyen es ESTIMATE Professional, SoftCost, entre otros.

II.8.3.2. Herramientas de hojas de tiempo

Las herramientas de hojas de tiempo apoyan el registro y reporte del esfuerzo en personas horas por actividad. Son una forma efectiva de recolectar el esfuerzo por actividad, siempre y cuando se registre el tiempo preciso que fue utilizado. Ejemplos son TimeSheet Professional para Windows de Timeslips Corporation y el shareware Timex.

II.8.3.3. Herramientas de calendarización del proyecto

Las herramientas de calendarización del proyecto apoyan la recolección de los datos respecto al calendario, esfuerzo y costo. Frecuentemente interactúa con hojas de cálculo para la captura de datos. Existen más de 100 herramientas. Algunos ejemplos son CA-SuperProject, Harvard Project Manager, MaCproject II, MicroPlanner, Microsoft Project.

II.8.3.4. Herramientas de reporte de status del proyecto

Los datos incluidos referidos inicialmente en un reporte de status del proyecto incluye el número de horas trabajadas, las fechas de inicio y fin de cada actividad, el número de unidades requeridas/completadas para cada actividad, y el número de unidades inspeccionadas. Las herramientas van desde simples editores de texto que usan plantillas hasta sofisticados paquetes de formularios. La mayoría de los editores, procesadores de palabras o bases de datos pueden producir reportes de status del proyecto. Si se cuenta con la capacidad para enviar dichos reportes por correo electrónico, esto se convierte en un valioso recurso.

II.8.4. Utilidades de apoyo

El siguiente grupo de herramientas no están enfocadas directamente a la medición, son herramientas genéricas que sirven para llevar control de las entidades de software, sin embargo, también auxilian en el uso de las métricas de software.

II.8.4.1. Herramientas de administración de la configuración

Las herramientas de la administración de la configuración (AC) apoyan al control de versiones y cambios. Las herramientas de administración de la configuración generalmente producen listas de todas las unidades de software controlada por la herramienta y su estatus (tal como inspeccionado o completado). Los aspectos de control de cambios de las herramientas de la AC apoyan cuando los requerimientos se agregan, cambian o borran. Ejemplos incluyen CCC/Manager de Softtool CVorp., PVCS de Intersolve, SourceSafe de One Tree.

II.8.4.2. Herramientas de base de datos

Se pueden utilizar herramientas de base de datos para establecer rápidamente las formas y reportes para registrar y reportar los defectos. Ejemplos de estas son ACCESS, FoxPro, dBASE IV.

II.8.4.3. Hojas de cálculo

Las hojas de cálculo agrupan los datos en simples renglones y columnas. Se pueden usar cuando se recolectan datos de la estimación y financieros. También se pueden usar efectivamente para reportes de defectos. Ejemplos conocidos son Excel, Lotus 123 y Quattro Pro.

II.9. Herramientas principales

De los tipos de herramientas, a continuación se describen y discuten las tres que más se aproximan al trabajo de investigación.

II.9.1. DataDrill

DataDrill [DataDrill] permite liberar datos críticos a administradores y tomar decisiones. Permite la definición de métricas según los objetivos de la organización. Cada usuario puede crear la representación que mejor se adapta a sus responsabilidades de administración. La información puede ser multiniveles y esta en función del individuo, se puede suscribir y copiar productos de información. Se puede tener la representación oficial de departamentos o programas. Esta herramienta provee las funciones necesarias para definir, implementar, y usar cualquier número de técnicas de administración y de medición. Las medidas y productos de información se despliegan en una gran variedad de gráficas y estilos. En DataDrill se importan diferentes cantidades y tipos de datos.

La herramienta brinda apoyo al trabajo colaborativo que pueden tener los equipos de trabajo durante el proceso de medición.

II.9.2. PSM Insight Version 4.2.2

Esta herramienta automatiza el proceso PSM. PSM Insight incluye las métricas que deben adaptarse, entrada de datos y funciones de análisis que ayudan a desarrollar una base de datos de mediciones del proyecto y análisis de estas. También PSM Insight provee plantillas de aspectos del proyecto y mediciones que comúnmente se usan, al igual que DataDrill también es completamente flexible para que se personalicen el análisis según las necesidades del proyecto.

II.9.3. TSPi Guide

TSPi Guide es una herramienta basada en Excel que le brinda apoyo automatizado a TSP (desarrollo del trabajo por equipos). Todos los formularios de TSPi Guide brindan apoyo total al proceso TSP. Con la información capturada, es posible generar todos los tipos de reportes que requieren ser analizados por TSP. En la Tabla I se comparan las principales cualidades que nos interesan conocer.

Tabla I.Comparación de las herramientas principales

Características	DataDrill	TSPi Guide	PSM Insight
Aplicación no colocalizada	✓		
Apoyo a equipos de trabajo	✓	✓	
Multiplataforma	✓		
Define métricas del administrador		✓	✓
Cubre el proceso de medición	✓	✓	✓
Memoria organizacional	✓		✓

Como se vio, hay una amplia variedad de herramientas que son muy particulares a elementos del desarrollo de software, herramientas que cuantifican líneas de código, que permiten el seguimiento de defectos o que contabilizan los requerimientos, entre otras, son muy comunes actualmente.

Por otro lado, a medida que la administración de proyectos ha ido cobrando importancia, también han ido apareciendo las herramientas de medición que auxilian al administrador para el control del proyecto. El apoyo esta dado principalmente por una amplia variedad las herramientas de estimación de proyectos.

Antes de describir la Tabla I, conviene señalar la importancia de los equipos de trabajo en los desarrollos de software actuales. La participación de los equipos de trabajo es algo natural a medida que los proyectos crecen, esta participación crea la comunicación y las dependencias entre los miembros del equipo. En un equipo de trabajo se hacen juntas (semanalmente, es un buen indicador) y las decisiones son tomadas después de largas discusiones donde se concilian varios criterios y es inherente la participación de los integrantes de los equipos.

De la Tabla 1, hay varias características a destacar. Cuando se habla de equipos de trabajo se refiere a que permitan las altas, bajas y modificaciones del equipo (como entidad) y de los ingenieros de software que integran el equipo. La presencia del líder es fundamental, es la cabeza del equipo y es el encargado de reportar y definir las métricas del equipo así como también discutir con los integrantes las causas de los problemas, entre otras muchas discusiones en las que puede estar envuelto.

El objetivo de TSPi es proveer una estructura de procesos para el trabajo en equipos. Estos procesos se definen en términos de la participación del equipo dentro del proceso de desarrollo de software. La relación entre las mediciones y los equipos esta dada por métricas que hablan del desempeño del equipo. Por otro lado, DataDrill brinda una perspectiva limitada del trabajo en equipo, la participación de los equipos en DataDrill ocurre cuando se asignan responsabilidades para la toma de decisiones y se trabaja en equipo para la creación de graficas. Finalmente, PSMi se enfoca meramente a una vista según el administrador de proyecto, el no manejar equipos de trabajo significa tener una visión muy pobre de las razones reales de porque se presentan los problemas, mas aún, cuando los equipos están distribuidos.

El término multiplataforma significa que la herramienta pueda ser ejecutada desde cualquier sistema operativo. DataDrill es una aplicación web y es la única herramienta que cumple esta característica, de hecho al correr sobre Internet, también es una herramienta distribuida que se puede acceder desde cualquier computadora.

La definición de métricas es una necesidad en los programas de medición actuales, la ventaja esta en que pueden definirse métricas sencillas pero significativas, métricas que estén en función de las necesidades de la administración. Las tres herramientas le dan soporte a esta característica.

El proceso de medición es algo que le dan soporte las tres herramientas y parcialmente lo cubren, básicamente estas herramientas cubren con las necesidades de recolectar información y presentarla.

La memoria organizacional ayuda especialmente a las estimaciones, DataDrill lleva un registro de todas las acciones que se han llevado a cabo. DataDrill y PSM Insight proveen una memoria para capturar la información actual en base a valores anteriores.

De estas tres herramientas, DataDrill es la herramienta mas completa, sin embargo su énfasis en las tarjetas Balance Score Card hace difícil ligarla únicamente a las necesidades de la administración de proyectos, las tarjetas Balance Score Card, principalmente resuelven necesidades de información de toda una empresa, aún así DataDrill provee apoyo a diferentes roles tal como puede ser un administrador de proyectos o un analista de sistemas. Por último, la interacción con otros miembros del equipo solo ocurre en la fase donde se preparan los monitoreos.

Después de todo el análisis de las herramientas, se tiene que son muy generales, o bien, muy particulares, al hablar de generales la pregunta es si esta generalidad es suficiente para cubrir con las necesidades de un administrador que trabaja con equipos de trabajo distribuidos. La respuesta es que ninguna herramienta esta enfocada a cubrir esta necesidad, es mas, cubren el proceso de medición de una manera arbitraria y la interacción de los equipos de trabajo con dicho proceso se presenta a lo más en una etapa del proceso de medición.

Lo que se propone en este trabajo de investigación es una guía que defina las características que debe tener un ambiente de medición para equipos de trabajo distribuidos, vistos desde el punto de vista de la administración de proyectos. En el siguiente capítulo se define las principales características y en el capítulo IV se reflejan estos conceptos a través del análisis y diseño de una herramienta que demuestre como se pueden implementar esta guía.

Capítulo III.

Proceso de medición para equipos de trabajo distribuidos

III.1. *Introducción*

Para la definición de esta guía fue necesario definir un proceso que especifique los elementos que intervienen para que un administrador de proyectos lleve control de proyectos de software con equipos de trabajo distribuidos.

Para la elección de estos elementos se tuvo en cuenta las limitaciones de los enfoques actuales y tomando en cuenta la importancia señalada para el contexto de las métricas que relata Fenton [Fenton y Neil, 2000], se busco definir un proceso donde el contexto de las métricas este definido cuando estas se adaptan al monitoreo de equipos de trabajo. En la Figura 2 se presentan los elementos del proceso SARE.

Inicialmente un administrador de proyectos esta a cargo de diferentes equipos de trabajo, tanto el administrador como los equipos de trabajo participan en el proceso de desarrollo de software, en dicho proceso los ingenieros mantienen una interacción constante con los elementos de la IS; la forma de trabajar es por ciclos o iteraciones, es decir, dividir el proyecto de software en diferentes iteraciones donde en cada ciclo se lleva a cabo el proceso de desarrollo y se tiene una versión de parte de la funcionalidad del sistema.

Cuando se quiere una mejora en los proceso de desarrollo, entonces los administradores necesitan conocer cuantitativamente el desarrollo del proyecto. La mejora del proceso se da en función de cubrir las necesidades de información y traducir estas a métricas de software, esto se conoce como la formulación de las métricas y corresponde a la primera fase del proceso de medición, una vez que se decide que medir, se recolectan los datos que se requieren, se grafican estos, se interpretan posibles anomalías y se da una retroalimentación para conocer en donde puede ocurrir una mejora dentro del proceso de desarrollo.

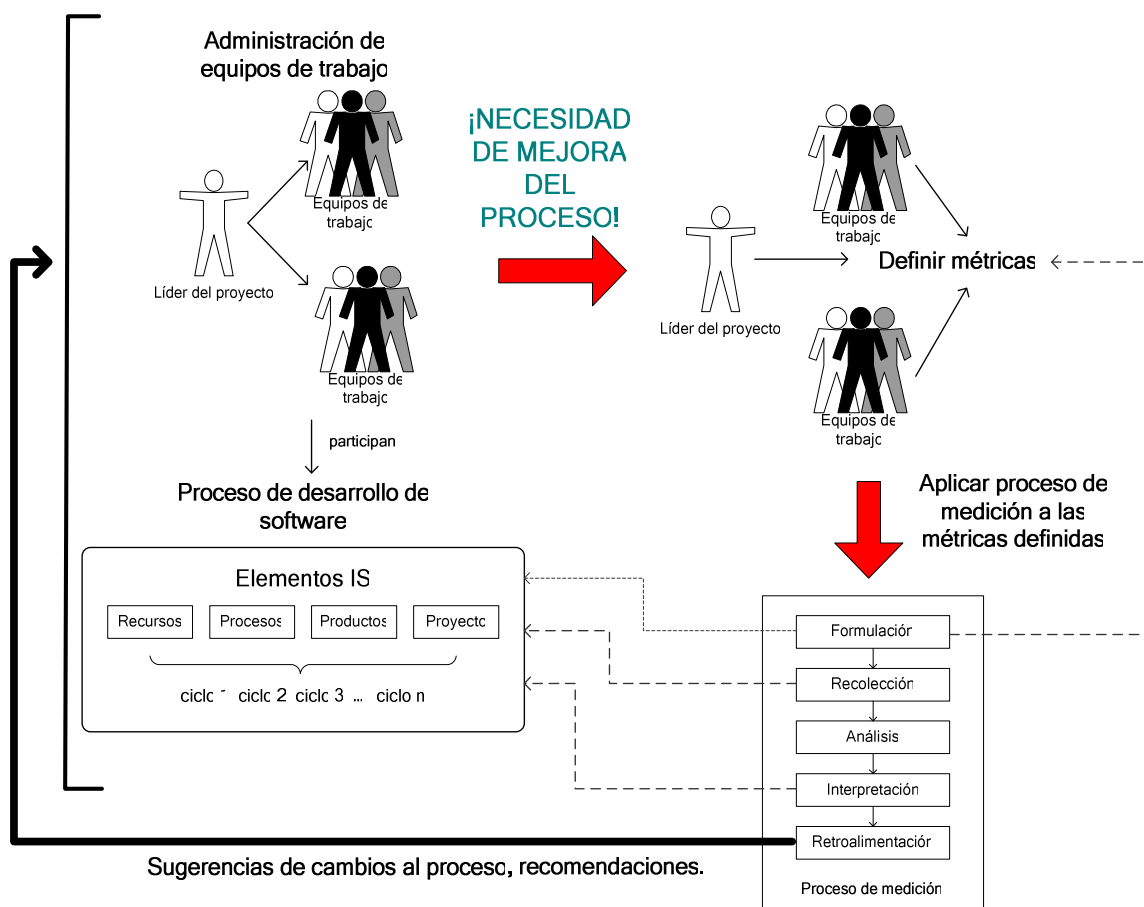


Figura 2. Proceso de medición para equipos de trabajo distribuidos.

La idea es aplicar mediciones a los elementos de la ingeniería de software y promover la participación de los equipos de trabajo tanto en el proceso de desarrollo como en el proceso de medición, todo esto, bajo la perspectiva de un administrador de proyectos.

Para la definición de este proceso se optó por agrupar las mejores prácticas y darles un valor agregado al unificarlas, siendo esta una práctica común en la ingeniería de software. Las mejores prácticas que se unifican son: (1) actividades detalladas del proceso de medición, (2) marco de desarrollo de software según TSPI, (3) elementos medibles de la ingeniería de software.

Con estas mejores prácticas, se incrementan las relaciones explícitas entre el contexto de desarrollo y las mediciones, se alcanza mayor visibilidad sobre el proceso de desarrollo y se garantiza así una aplicación efectiva de las métricas.

En la siguiente sección se describe con mayor detalle cada una de estas mejores prácticas aplicadas al trabajo de investigación.

III.2. *Prácticas enfocadas a la guía*

En esta sección se describen en detalle que elementos se consideraron a partir de las mejores prácticas, para el criterio de la elección se considero principalmente aquella información que apoye al proceso de medición.

III.2.1. Proceso de medición para equipos distribuidos

Un ingeniero estima una medición según el método que mas le convenga (experiencia personal, analogía, modelo) y calendariza cuando se calculará el valor actual o bien se hará una nueva reestimación. Después con los monitoreos y los periodos de tiempo en que se revisaran estos para conocer la evolución de dicha métrica. Lo siguiente es establecer cuando se revisarán las gráficas y/o reportes, es decir, cuando serán interpretadas por los miembros del equipo y se dará retroalimentación al proceso de desarrollo. A continuación se describen cada una de las fases del proceso de medición adaptado con la participación de equipos de trabajo distribuidos.

III.2.1.1. Formulación

Se implementan métricas a nivel de la administración del proyecto tales como el esfuerzo y duración, estas se seleccionan de un catálogo y solo pueden ser formuladas por el administrador o líder del equipo. La guía debe ser flexible para definir métricas, de acuerdo a la necesidad del administrador de proyecto, analista, diseñador, programador o ingeniero de pruebas.

III.2.1.2. Recolección

La recolección se realiza según los roles, las mediciones se recolectan dependiendo de la etapa del proceso de desarrollo en las que se encuentren, no se trata de recolectar sin orden, más bien dependiendo de la fase de desarrollo, es cuando resulta útil estimar o calcular el valor que se presenta en una medición. Se puede calendarizar los eventos o periodos de tiempo de cuando se recolectarán las métricas, por ejemplo puede ser semanalmente, después de cada ciclo o bien en el postmorten del proyecto de desarrollo.

III.2.1.3. Análisis

Se programan los monitoreos (gráficas y reportes) que deben estar listos para una fecha o evento particular. Los monitoreos semanales son una buena medida de cuando hacer los monitoreos. La precondition para hacer un monitoreo, es que este asociado a una estimación. Se debe cuidar que los monitoreos que se establezcan sean significativos, es decir, que haya altas probabilidades de que brinden información relevante. Finalmente, cualquier ingeniero puede programar monitoreos según las métricas que es responsable, sin embargo, lo más común es que el líder del equipo sea el encargado de programar los monitoreos.

III.2.1.4. Interpretación

Para la interpretación de los monitoreos, es posible contar con sugerencias [Kitchenham y Pickard, 1987] (basadas en la literatura, memoria organizacional) que expliquen el comportamiento de las gráficas o los resultados de los reportes. Cada ingeniero puede ver los monitoreos de sus compañeros de equipo y dar opiniones.

La información esta dividida por módulos, subsistemas y sistemas, la idea es interpretar lo que dice la métrica en un contexto particular y de ser necesario se debe contar con mecanismos para revisar la información de otros contextos de desarrollo, o bien explorar el comportamiento de otras métricas asociadas. Al contemplar solo las métricas de la administración del proyecto, se evita que se haga una revisión exhaustiva de los monitoreos (dada sus métricas asociadas), sin embargo, ya que todo es un proceso de aprendizaje de causas y de contextos de desarrollo, desafortunadamente también se tienen pérdidas de información; por ejemplo, no es posible regresar a capturar datos que resultaba interesante el haberlos considerado, pero lo que si es posible es registrar este hecho.

El resultado en esta fase es el tener identificadas las causas que provocan los efectos anómalos de las mediciones.

III.2.1.5. Retroalimentación

En esta fase se plantean soluciones a las causas que se descubrieron en la interpretación. También se indican las métricas y los responsables de estas, para la que se espera que reflejen una mejoría, generalmente estas son las métricas del próximo ciclo de desarrollo, sin embargo, puede programarse para antes del fin del ciclo, en ambos casos inicia de nuevo el proceso de medición. Esta fase usualmente concluye con el fin del ciclo de desarrollo, aunque también pudo haber terminado antes de que concluya el ciclo. La retroalimentación es el resultado de todo el proceso de medición y su eficiencia depende de como funcionó este.

III.2.2. Consideraciones de TSPi

Para el propósito del trabajo de investigación, un contexto de desarrollo debe tener guías definidas que cubran todo el proceso de desarrollo, además este contexto debe apoyar el trabajo en equipos. La elección de TSPi se presenta como la mejor opción dada la claridad de sus procesos enfocados a equipos de trabajo.

TSP es una guía extensa que describe en detalle las características que se siguen cuando se hace un desarrollo de software por equipos de trabajo, sin embargo, no es necesario considerar todas estas características para el trabajo de investigación, solo aquellas, que brinden un contexto para los elementos de IS y para las métricas de software, a continuación se describen estas características.

- *Desarrollo por ciclos.* El desarrollo por ciclos de trabajo divide al proceso de desarrollo en diferentes ciclos, donde en cada ciclo de trabajo se plantean objetivos

que alcanzar. Una vez que se hace una planeación de los ciclos (donde en cada ciclo usualmente se tiene un entregable), el equipo trabaja enfocado a cumplir con lo planeado en el ciclo. El trabajo por ciclos da contexto para saber que requerimientos necesitan completarse en determinado ciclo, así como por ejemplo, el esfuerzo que se requiere en cada ciclo.

- *Proceso guiado por requerimientos.* Para tener una idea clara del desarrollo del proyecto, de saber si se terminara o no a tiempo o de saber donde se han tenido problemas, los administradores tienen la necesidad de un proceso guiado por los requerimientos, el poder trazar los requerimientos por las diferentes etapas del desarrollo, ayuda a tener una visión de la factibilidad de éxito en el desarrollo.

III.2.3. Equipos de trabajo

Un equipo de trabajo requiere de varios elementos para que operen de manera coordinada. Según Morisio un proceso de medición es un proceso de desarrollo, a su vez los procesos de desarrollo presentan colaboraciones entre los miembros del equipo [Morisio, 1999]. Loomis hace referencia a un estudio de los factores más importantes para mantener colaboraciones exitosas en el desarrollo de software [Loomis, 1997], a continuación se describen estos factores aplicados al trabajo de investigación:

- *Liderazgo.* El líder es el encargado, quien merece el respeto del equipo y quien designa responsabilidades y dueños de las métricas.
- *Metas claras.* Con una meta clara, se sabe hacia donde gira toda la información, esta debe ser perfectamente visible y enfocar el esfuerzo del equipo para alcanzarla. La meta común surge a través de una necesidad de información y esta es dada por una métrica en particular
- *Roles.* Los roles permiten evitar la incertidumbre de lo que tiene que hacer cada ingeniero. Este tiene que ver con el principio *responsabilidades compartidas* definido anteriormente.
- *Calendario.* Un calendario sirve para que el equipo conozca cuando se tienen que hacer los entregables, esto se aplica teniendo tiempos de entrega razonables para cualquier producto del proceso de medición, así como las actividades de este.
- *Interdependencias.* Las interdependencias ayudan básicamente a conocer el impacto de los cambios, para esto se guardan interdependencias entre elementos del proceso de desarrollo y del proceso de medición, esto facilita conocer las repercusiones de algún cambio.
- *Proceso de resolución.* Es tener procesos definidos que ayuden a descubrir la mayoría de los problemas y solucionarlos, para cumplir esto, se cuentan con fases del proceso de medición sobre las cuales se llevan a cabo diversas discusiones.
- *Soporte por comunicaciones.* Se refiere a tener un sistema de comunicación bien diseñado, para esto se tiene libertad para establecer juntas cada determinado espacio de tiempo, interactuar por correos electrónicos o boletines de discusión.

III.2.4. Métricas para los elementos de la IS

El proceso de desarrollo de software no es una línea de producción donde siempre es posible conocer que engrane es el que produce ciertas inconsistencias, más bien, es un proceso intelectual donde son muchos los factores que pueden producir cambios, estos factores se asocian a los elementos de la IS.

Para establecer una relación explícita con el proceso de desarrollo de software, se definen las métricas según los elementos de la IS, es decir, métricas del proyecto, productos, procesos y recursos (tal como se vio en el capítulo II),

Para cada uno de los elementos de la IS, se utilizan las métricas de PSM gracias a su simplicidad y a que constituyen el estándar de facto.

III.2.4.1. Proyecto

En la literatura, las métricas del proyecto son de las más estudiadas, de hecho estas métricas corresponden a lo que se conoce como la planeación de proyectos de software.

La principal proliferación se debe al modelo de estimación de costos COCOMO de Barry Boehm [Boehm, 1981], la primera versión de este modelo fue COCOMO 81, el estándar actual es COCOMO II.

El modelo COCOMO a través de una serie de fórmulas produce estimaciones del tamaño, esfuerzo y duración del proyecto. Las fórmulas se ven afectadas por una serie de parámetros aplicables a cualquier proyecto, por ejemplo, fase del desarrollo en la que se encuentra el desarrollo o nivel de experiencia de los ingenieros de software, entre otros parámetros.

También es común utilizar la descomposición por módulos donde se aplican las métricas del proyecto a cada módulo y después la suma total de estos valores dan una aproximación de la métrica aplicada a todo el proyecto.

III.2.4.1.1. Tamaño

Se especifica el tamaño según las líneas de código, puntos de función o puntos de objeto, estas tres unidades dan mayor precisión según la etapa del desarrollo de software en la que se hace la estimación.

- *Puntos de objeto (PO)*. Cuando no se han definido aún los requerimientos formalmente pero se tiene una idea de lo que será el sistema y es factible visualizar las pantallas, informes y tablas de la base de datos, entonces se cuantifica en términos de los puntos objeto. Los PO se obtienen a través del modelo de COCOMO.

- *Puntos de función (PF)*. Una vez que el documento de requerimientos ha sido definido, entonces la estimación por puntos de función se obtiene a través de elementos contextuales que influyen en el desarrollo del producto, también se involucran factores como las entradas y salidas del sistema a desarrollar, entre otros parámetros. Los PF se obtienen mediante el modelo de COCOMO.
- *Líneas de código (LOC)*. La medida en LOC o en miles de líneas de código (KSLOC) es una de las más arraigadas desde que se ha empezado a cuantificar el software. Se contabilizan las líneas de código no comentadas. A medida que se avanza en el desarrollo y se tienen más módulos implementados, entonces se tienen estimaciones más precisas.

III.2.4.1.2. Esfuerzo

Son el número de las personas por mes que participan en el proyecto, con el modelo de COCOMO se puede obtener una estimación del esfuerzo una vez que se ha estimado el tamaño del proyecto.

III.2.4.1.3. Duración del proyecto

La duración del proyecto generalmente se divide en meses, sin embargo la duración de cada ciclo de desarrollo es de unas siete semanas aproximadamente [Humphrey, 1999]. Con el modelo de COCOMO se obtiene esta estimación una vez que se obtuvo el esfuerzo necesario para el proyecto.

III.2.4.1.4. Costo

El costo del proyecto está en términos de la cantidad económica que será necesario invertir en el proyecto. Depende del salario que percibirá cada ingeniero y de los costos de los componentes externos de terceros (COTS).

III.2.4.2. Recursos

Son los recursos humanos que se tienen disponibles y la medida en que participan en el proyecto. También se toman en cuenta los componentes de software que se utilizan.

III.2.4.2.1. Personal

Es el número de participantes del proyecto. Se consideran las horas de trabajo y las horas extras invertidas no solo en el proyecto, también en cada fase de desarrollo y por consiguiente en cada iteración y productos que se generan. También se contabiliza el número de proyectos en los que un ingeniero trabaja.

III.2.4.2.2. Experiencia

Es el nivel de entrenamiento que ha recibido el equipo, se registra la fuente de donde obtuvo experiencia, por ejemplo, a través del conocimiento empírico (autodidacta o por medio de un compañero) o bien por un curso de capacitación.

III.2.4.2.3. Productividad

Es el número de productos de software por unidades de tiempo que desarrolla cada ingeniero, por ejemplo, puede ser, número de requerimientos definidos por día, o bien KSLOC por semana.

III.2.4.2.4. Componentes de software reutilizables

Son los componentes que se reutilizan, ya sea que fueron desarrollados dentro de la organización o bien que fueron adquiridos por terceros.

III.2.4.3. Producto

Los productos que se generan son requerimientos, diseño, código fuente, y el plan de pruebas. Los requerimientos y el plan de pruebas son documentos que comúnmente no cambian según el desarrollo del software, sin embargo, el diseño y la implementación varían según el paradigma de programación que los genera.

III.2.4.3.1. Tamaño

Se contabilizan atributos de los productos que se generan en cada fase, estos son:

- *Análisis*. Número de requerimientos funcionales y no funcionales
- *Diseño*. Número de módulos.
- *Implementación*. Miles de líneas de código implementadas (KSLOC).
- *Pruebas*: Número de casos de prueba.

También se registra el tipo de incremento, es decir, los requerimientos, módulos o líneas de código que son nuevos o bien que se actualizaron.

III.2.4.3.2. Calidad

La calidad del producto se traduce a defectos y un defecto ocurre cuando un error humano resulta en algún error dentro del producto de software, el defecto es identificado por el desarrollador. Los tipos de defectos dependen del producto.

- *Requerimientos*. En la literatura se tienen muchos estudios que parten del trabajo de Davis [Davis et al, 1993], entre estos se encuentra las métricas de calidad de la IEEE. Del estudio de Davis, las métricas más sobresalientes son:
 - *Correctitud*. Cada requerimiento representa algo que el sistema requiera construir.
 - *Ambigüedad*. Cada requerimiento tiene una sola posible interpretación.
 - *Completez*. Un requerimiento es completo si:
 - Todo lo que el sistema se supone que hace esta descrito en la Especificación de Requerimientos del Software (SRS, por sus siglas en inglés).
 - Es la respuesta del software a todos los datos de entrada en todas las posibles situaciones.

- Todas las páginas, figuras y tablas están numeradas, etiquetadas y referenciadas, todas las unidades de medición se proveen y se presenta el material de referencia.
 - No se cuentan con secciones marcadas “a ser determinada”.
- Consistencia. Ninguno de los requerimientos escritos entra en conflicto.
- Ordenado: Por importancia, estabilidad o versión. Los requerimientos tienen identificadores que indican su importancia, estabilidad o versión.
- Verificable. Existe un proceso efectivo de costos finitos en la que una persona o una máquina puede verificar que el producto de software satisface al requerimiento.
- Modificable. El estilo y la estructura son tales que cualquier cambio al requerimiento puede ser hecho fácilmente, completamente y consistentemente, al tiempo que se mantiene la estructura y el estilo.
- Rastreadable. El origen de cada requerimiento es claro y es fácil referenciar cada requerimiento durante desarrollos futuros. Es decir, cada requerimiento que tiene un origen, entonces es referenciado a dicho origen.
- Entendible. Todos los tipos de lectores comprenden el significado de los requerimientos con una mínima explicación.
- Alcanzable. Existe al menos un diseño del sistema e implementación que lo implementa.
- Conciso. Es tan corto como sea posible sin afectar su calidad.
- Independencia del diseño. Existe más de un diseño e implementación que correctamente lo implementa.
- No redundante. El requerimiento no se ha establecido más de una vez en la SRS.
- Preciso. Se cumple esta propiedad cuando:
 - Se usan cantidades numéricas cuando es posible.
 - Se usan los niveles apropiados para todas las cantidades.
- Reusable. Las oraciones, párrafos y secciones pueden ser adaptados para subsiguientes SRS.
- Mantiene referencias cruzadas. Establece referencias a:
 - Requerimientos idénticos.
 - Descripciones más abstractas de los requerimientos.
 - Requerimientos que dependen de dicho requerimiento, o bien, las referencias de los requerimientos de los que depende.
- *Diseño e implementación.* Los defectos de diseño (bajo y alto nivel) e implementación se basan en la clasificación ortogonal de defectos de Chillarege [Chillarege et al, 1992]. Esta clasificación es la base de los defectos del Proceso de Software Personal (PSP por sus siglas en inglés) [Humphrey, 1995] y en consecuencia para TSP. La importancia de esta taxonomía radica en que es útil para detectar las causas de los defectos y la efectividad de los métodos para hallar los defectos. A continuación se presentan los tipos de defectos y donde es que estos se ven reflejados.
 - Documentación. Comentarios, mensajes.
 - Sintaxis. Ortografía, puntuación, tipos, formato de las instrucciones.

- Construcción, empaquetamiento. Administración de los cambios, bibliotecas, control de versiones.
- Asignación. Declaración, nombres duplicados, alcance, límites.
- Interface. Llamadas y referencias a procedimientos, Entrada/Salida y formatos del usuario.
- Chequeo. Mensajes de error, chequeos inadecuados.
- Datos. Estructura, contenido.
- Función. Lógica, apuntadores, ciclos, recursiones, cálculos, defectos de la función.
- Sistema. Configuración, memoria.
- Ambiente. Diseño, compilación, pruebas.
- *Número total de defectos.* Este se reporta por requerimiento y por módulo, así como para el número total de requerimientos o módulos.
- *Costo.* Es el tiempo que se invierte para corregir un defecto. Corregir defecto genera que no haya productividad.
- *Severidad del defecto.* Es el impacto que causa el defecto y este puede ser menor o mayor. Si el defecto es menor se puede corregir hasta que inicie la depuración y si es mayor inmediatamente debe ser corregido.
- *Edad del defecto.* Se refiere al intervalo de tiempo en que el defecto se detecto y corrigió.

III.2.4.3.3. Estabilidad

Son las solicitudes de cambios al producto, se registra:

- El número de solicitudes de cambios que se generan.
- Número de cambios que se generan.
- *Tipo de cambio.* Este puede ser menor o mayor. Es menor si no esta asociado a referencias cruzadas y mayor cuando si lo esta.

III.2.4.3.4. Estado

El estado corresponde a la pregunta, ¿esta listo el producto para ser entregado?, los estados son: listo, en progreso y para este último se registra el porcentaje que se ha alcanzado.

III.2.4.3.5. Casos de prueba

Los casos de prueba se aplican sobre los productos de software. Estos generan los siguientes estados: en ejecución, completado o ejecutado, no completado o pendiente.

III.2.4.3.6. Fase del desarrollo

Cada producto de una fase mantiene una representación con un producto de otra fase.

- Requerimientos que están siendo diseñados, implementados, o en etapa de pruebas
- Módulos de diseño que han sido implementados o están en pruebas.
- Módulos implementados que están en pruebas.

III.2.4.4. Proceso

Las métricas del proceso son sobre las fases de análisis, diseño, implementación y pruebas. Los procesos de revisión, inspección y pruebas sirven para conocer la efectividad de cada una de estas fases.

III.2.4.4.1. Duración de cada fase

Es el tiempo invertido para completar una fase. Pueden ser en días, semanas o meses.

III.2.4.4.2. Esfuerzo

Son el número de ingenieros que participan en la fase (ver métricas de recursos).

III.2.4.4.3. Calidad

La calidad impacta la efectividad con que se ejecuta una fase. Los defectos se detectan gracias a las revisiones, inspecciones o pruebas, las métricas que se registran son:

- Número y tipo de defectos que se detectan en la fase (pueden ser defectos introducidos en la fase actual o fases previas).
- Número y tipo de defectos que son removidos en la fase.
- Número de defectos que pasan inadvertidos.

Ya se han descrito los detalles de las mejores prácticas que llevan a la definición de la guía, en la Figura 3 se presenta una gráfica rica que representa el proceso de medición para equipos de trabajo distribuidos.

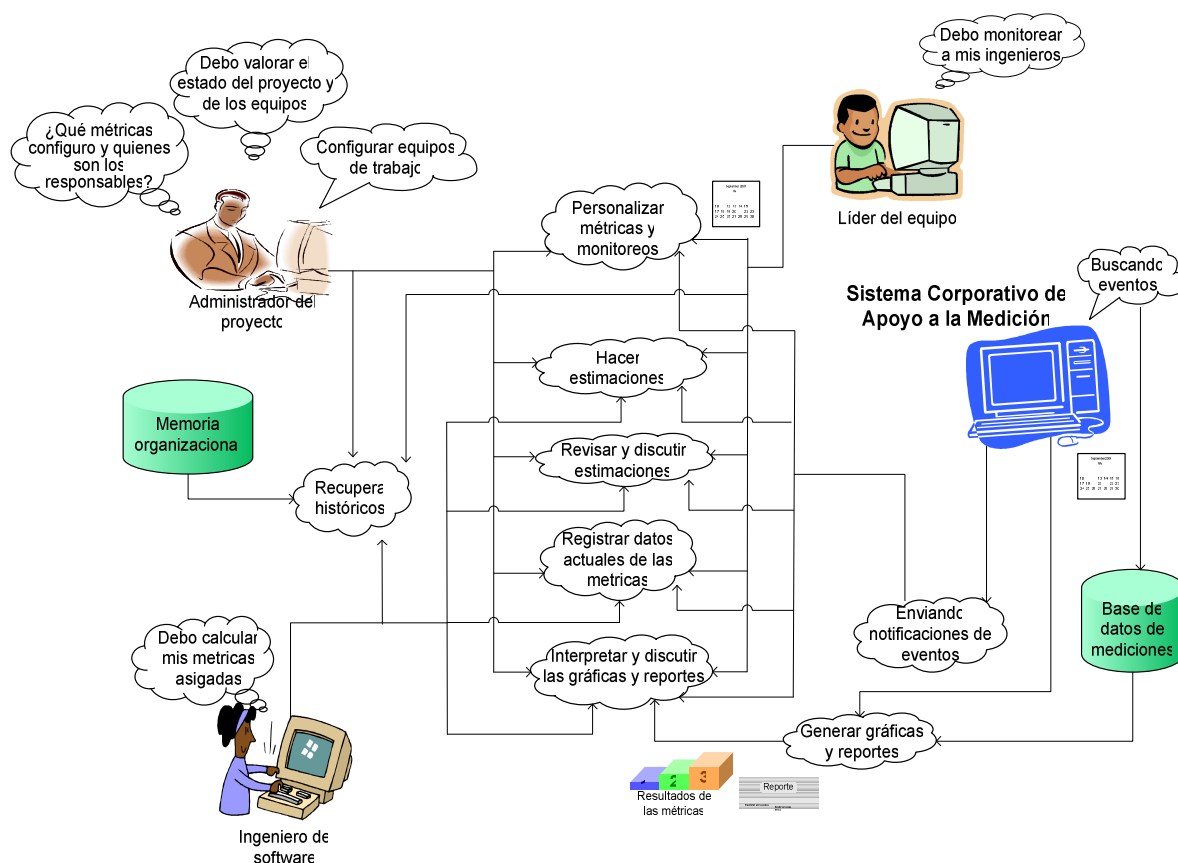


Figura 3. Gráfica rica del proceso de medición orientado a equipos de trabajo.

En la Figura 3 se presentan los roles que participan en el proceso SARE, esta el líder del proyecto, el líder del equipo de trabajo y el ingeniero de software, estos usuarios tienen diferentes necesidades, por ejemplo, el líder tiene la necesidad de crear equipos de trabajo y monitorearlos a través de métricas que sean de su interés, el líder define mediciones según el equipo que controla y finalmente el ingeniero puede registrar mediciones referente a su trabajo individual.

Los usuarios de SARE dan un seguimiento detallado de las métricas a través del proceso de medición. Para tener una mayor riqueza y retroalimentación en las métricas, los ingenieros discuten en las fases de recolección, análisis, interpretación y retroalimentación. También gracias a los datos históricos (memoria organizacional) se da mayor sentido y certeza a la información actual.

Una vez que se han definido las fuentes de información (experiencia de miembros del equipo, datos históricos) que ayudan a contextualizar a las métricas, ahora es necesario contar con la información a tiempo y para esto se hace uso del Sistema Corporativo de Apoyo a la Medición. Este sistema lleva un registro de las mediciones y de las acciones que

se han aplicado y las que están por aplicarse sobre las métricas, gracias a esta información se envían notificaciones a los usuarios y se alcanza el objetivo de contar con la información a tiempo. En la Figura 3 aparecen dos componentes de información, a continuación se describen estos.

III.3. Componentes de información

Para el proceso de medición adaptado a equipos de trabajo distribuidos, se incorporan dos componentes de información principal, la base de datos de mediciones y la memoria organizacional.

- *Base de datos de mediciones.* Es un repositorio donde se almacena la información de las mediciones, cada métrica que se registra tiene asociado una versión que sirve para mantener un control de la configuración. La medición es una entidad muy activa que mantiene un estado y relaciones con entidades del proceso de desarrollo, por ejemplo, una medición puede ser estimada, capturada, monitoreada, entre otros estados, también una medición esta relacionada a un responsable y varios ingenieros pueden participar en una estimación grupal sobre una medición dada, también hay eventos que activan acciones sobre una medición tal como puede ser un monitoreo. Todas estas y otras propiedades están reflejadas en la base de datos de métricas.
- *Memoria organizacional.* No hay métricas sin datos históricos, esta información sirve para tomar decisiones presentes y hacer estimaciones futuras. Debe ser de fácil acceso según las necesidades de información de los ingenieros.

Una vez que se tienen los componentes y el detalle de las mejores prácticas que se consideran, entonces se pueden presentar los principios para la implementación de la guía, esta se describe a continuación.

III.4. Principios

A continuación se describen los conceptos que se requieren para implementar las métricas de software de la administración de proyectos para equipos de trabajo que están distribuidos.

- *Administración de las mediciones.* Se lleva una administración efectiva de las mediciones a través del control de la configuración de estas, así como también se registra la interacción de las métricas con su contexto, las diferentes versiones de una medición y las versiones del producto o procesos de software asociados. A la larga, con toda esta información se tienen bancos de datos históricos. Solo es necesario administrar las mediciones y conocer los valores que se obtuvieron, actualmente se cuentan con muchas herramientas que tienen mecanismos para la obtención automática de estos valores.
- *Notificaciones.* Las notificaciones son elementos que suceden de acuerdo a reacciones y que hacen que la información viaje lo más rápido posible. Es necesario

tener la información disponible y verificar cual es el cambio y como impacta sobre lo que se esta produciendo. Por ejemplo, si un requerimiento no quedo claramente entendido, entonces puede afectar a tres módulos y esto a su vez a siete casos de prueba. El contenido y destinos de tales notificaciones dependen de la aplicación de las métricas, por lo tanto es necesario definir reglas para cada caso de aplicación. Por lo tanto, se necesita un sistema de notificaciones en la que un usuario establezca (personalice) dichas reglas.

- *Responsabilidades compartidas.* Las políticas y los propietarios de la información están en función del rol que les corresponde. Además del administrador de proyectos quien ve la información de los equipos de trabajo, están los líderes de equipo que ven la información de los ingenieros que tienen a su cargo, finalmente están ingenieros de requerimientos, de diseño, de implementación y de pruebas, quienes solo ven su propia información
- *Discusiones.* Las métricas son un ambiente de interpretación de resultados, la variedad de opiniones puede brindar una interpretación precisa solo si todas estas opiniones son bien encausadas y tomadas en cuenta.
- *Fomenta la participación de todos los integrantes del proyecto.* A medida que cada ingeniero tiene responsabilidades compartidas y participa en las discusiones, se alcanza un mayor aprendizaje grupal. Por otro lado, conforme se tiene más colaboración y retroalimentación se contará con un proceso de medición más efectivo y rico en información.
- *Configurabilidad sobre el uso de las métricas.* Se debe contar con mecanismos flexibles para poder tomar decisiones sobre que métricas monitorear, cuando capturarlas y evaluarlas. Al configurar gráficas y reportes, se busca agrupar la información de modo que se visualicen grandes cantidades de datos. Es muy importante en que el número de graficas que se presenten sean las necesarias para la toma de decisiones.
- *Aprovecha las experiencias de los miembros del equipo a través de la memoria organizacional.* Cada usuario tiene diferentes perspectivas, cada vez que se vaya avanzando en el desarrollo, se tendrán situaciones que probablemente ya se han presentado y fueron resueltas. Al final se tiene el registro de las causas de los problemas.
- *Provee una jerarquía de métricas para la administración de proyectos.* Contar con mediciones ya establecidas reduce las variantes de tener programas de medición complejos, donde es necesario establecer metas, preguntas y mediciones tal como sucede en GQM.

En el siguiente capitulo se describe el análisis, arquitectura y diseño de la herramienta SARE, el fin de esta herramienta es reflejar la implementación de esta guía.

Capítulo IV.

Análisis y diseño del sistema SARE

IV.1. Introducción

El análisis y diseño debe estar presente en cualquier desarrollo de software, estas etapas ayudan a tener una mayor certeza de si el producto final que se obtiene cumple las expectativas de los usuarios. En este capítulo se presenta el análisis y diseño del sistema SARE.

En el análisis se obtienen o descubren las necesidades de los usuarios finales traducidas a requerimientos de software. Al final del análisis se tiene una visión de “que” es lo que debe cumplir el sistema no del “como” es que lo cumple.

El diseño consiste en modelar los requerimientos obtenidos en la fase de análisis, los requerimientos se traducen a una representación de diseño que define como es que se cubre cada requerimiento.

IV.2. Análisis

En esta sección se describen los requerimientos de tipo funcional, de información y colaborativos que cubren en detalle las capacidades del sistema SARE. También se describen los diagramas de casos de uso y de secuencia que cubren la fase de análisis.

IV.2.1. Requerimientos

Pressman sugiere que el primer paso para identificar los requerimientos, es preguntar al cliente, a los usuarios y a los que están involucrados en los objetivos del sistema por sus necesidades o cómo el sistema se ajusta a las necesidades requeridas [Pressman, 2002].

Los requerimientos son la guía del proyecto, cualquier cambio sobre los requerimientos se traduce en costos sobre todo el proyecto, por lo tanto, una buena ingeniería de requerimientos es fundamental.

Una vez que se tienen los requerimientos, estos comúnmente se clasifican en: requerimientos de información y requerimientos funcionales, dada la naturaleza de este trabajo también se considera la clasificación de los requerimientos colaborativos. Unos requerimientos son útiles si son de calidad, es decir, si cumplen con las métricas de calidad de los requerimientos que fueron descritas en la sección III.5.3.2.

A continuación se describen los requerimientos del sistema SARE. Estos se organizan según las fases del proceso de medición y según el contexto administrativo para el PM, con este contexto se brinda mayor facilidad de interpretación a las métricas de software.

IV.2.1.1. Requerimientos de información

Los requerimientos de información son los datos que se necesitan o que se producen según los requerimientos funcionales para que estos se cumplan satisfactoriamente. A continuación se describen estos requerimientos.

IV.2.1.1.1. Formulación

- RIF1. *Catálogo de métricas.* Es un catálogo de métricas según los elementos de la ingeniería de software.
- RIF2. *Elemento donde se aplica.* Es elemento de la IS, estos son:
 - *Proyecto*
 - *Producto (documento de análisis, diseño, implementación (sistema) y plan de pruebas)*
 - *Procesos*
 - *Recursos*
- RIF3. *Ciclo.* Ciclo actual del desarrollo.
- RIF4. *Responsable de la métrica.* Es cualquier ingeniero de software, es conveniente que sea el ingeniero responsable del elemento sobre el que se aplica la métrica.
- RIF5. *Identificador de la métrica.* Cada métrica tiene asociado un identificador.

IV.2.1.1.2. Recolección

- RIR1. *Elemento donde se aplica.* Elemento de la IS (ver RIF2).
- RIR2. *Métodos/Modelo usados.* Es el tipo de método: experiencia personal, históricos o modelos (como lo es COCOMO II).
- RIR3. *Información histórica.* Son valores anteriores del proyecto actual, o bien de proyectos anteriores que sirven para la estimación.
- RIR4. *Versión de la estimación.* Es una referencia que registra a la estimación bajo un control de la configuración (control de versiones).
- RIR5. *Valor de la estimación.* Es el valor estimado en determinadas escalas de medición y que se ubica dentro de un límite alto y bajo.
- RIR6. *Fuente.* Es el medio (revisión, inspección, pruebas) por el que se obtiene el valor actual de la estimación.
- RIR7. *Suposiciones (parámetros críticos).* Son los valores de los parámetros de un modelo que se combinan para generar una estimación. Se pueden generar estimaciones de contingencia cuando se superan los valores de dichos parámetros.
- RIR8. *Racional de la reestimación.* Es la descripción del porque de la reestimación, se mantienen referencias a restricciones del proyecto descritas en planes de proyecto y planes individuales.
- RIR9. *Fecha de reestimación (fecha y hora).* Cuando debe estar lista la reestimación y cuando es que esta se captura.
- RIR10. *Eventos.* Son eventos que reactivan la reestimación.
- RIR11. *Ingeniero Responsable.* Es el ingeniero responsable de la métrica.
- RIR12. *Participantes.* Son los ingenieros responsables de discutir el valor de una reestimación.

RIR13. *Fecha de captura de métrica.* Hora y fecha para la recolección de los datos del valor actual.

IV.2.1.1.3. Análisis

- RIA1. *Fecha.* Es un periodo de tiempo para hacer un análisis.
- RIA2. *Eventos.* Son eventos que activan el análisis.
- RIA3. *Ingeniero responsable.* El ingeniero encargado de configurar los parámetros del análisis.
- RIA4. *Participantes.* Son los ingenieros encargados de discutir el análisis.
- RIA5. *Objetivo del análisis.* Este puede ser confirmar una teoría o explorar una relación.
- RIA6. *Monitoreos asociados.* Son aquellos monitoreos que buscan explicar un valor anómalo.
- RIA7. *Métricas asociadas.* Son las métricas que influyen a una métrica particular.
- RIA8. *Tipos de análisis.* Puede ser un análisis multivariado (la relación de diferentes métricas) o series de tiempo (una métrica a través del tiempo) y por componentes.
- RIA9. *Valores anómalos.* Son los valores que están fuera de lo estimado
- RIA10. *Información histórica.* Son valores anteriores del proyecto actual o bien de proyectos anteriores que sirven para comparar los valores anómalos.

IV.2.1.1.4. Interpretación

- RII1. *Ingeniero responsable.* Es el ingeniero encargado de configurar los parámetros de la interpretación.
- RII2. *Participantes.* Son los ingenieros que discuten con el fin de dar una interpretación a los monitoreos.
- RII3. *Fecha.* Es la fecha de inicio y en la que se tiene lista la interpretación del monitoreo.
- RII4. *Información histórica.* Son interpretaciones de los monitoreos del proyecto actual o bien de proyectos anteriores que sirven para conocer las causas actuales.
- RII5. *Lista de circunstancias.* Es la racional para conocer que representan las métricas, son causas de las excepciones que ocurren. Estas pueden generarse de acciones correctivas, adaptación, preventiva y perfectiva.
- RII6. *Tipo de interpretación que brinda.* Favorable (habla de un desarrollo sin dificultades y con prácticas de desarrollo de software efectivas), no favorable (se presentaron problemas en el desarrollo).
- RII7. *Información adicional.* Es información (puede ser histórica) que determina el tipo de interpretación del monitoreo.

IV.2.1.1.5. Retroalimentación

- RIRET1. *Participantes.* Son los ingenieros que participan en decidir cuales serán las acciones de administración según los resultados de la interpretación.
- RIRET2. *Fecha.* Es la fecha en que inicia y en la que se tiene listo el reporte de la retroalimentación.

- RIRET3. *Benefactores directos de la retroalimentación.* Son los ingenieros que se ven más afectados según la retroalimentación.
- RIRET4. *Listas de acciones de administración.* Son acciones según las causas de la interpretación y están destinadas a corregir los problemas que reflejan las métricas.
- RIRET5. *Métricas a monitorear.* Son banderas para las métricas de los productos sobre las que impacta la retroalimentación.
- RIRET6. *Información histórica.* Se mantiene una referencia para las listas de retroalimentación que no fueron exitosas así como las que si lo fueron.
- RIRET7. *Gravedad.* Es el impacto de las acciones de retroalimentación, pueden ser perfectivas (optimización) o correctivas (necesariamente deben ser corregidas).

IV.2.1.1.6. Administración

- RIADM1. *Equipos de trabajo.* Nombre del equipo, descripción (objetivo) del equipo, líder.
- RIADM2. *Ingenieros.*
- Nombre.
 - Identificador (nombre de usuario, clave), correo electrónico,
 - rol(es),
 - Equipo(s) al que pertenece(n).
- RIADM3. *Proyecto.*
- Nombre.
 - Identificador (clave).
 - Descripción.
 - Fecha de inicio.
 - Fecha de fin.
 - Número de ingenieros.
 - Capital con que se cuenta.
 - Líder del proyecto.
 - Tipo y subtipo del proyecto, estos pueden ser:
 - Sistema de negocios, control, internet, intranet, micro código, firmware, científico, investigación de ingeniería, software de sistema/controladores y telecomunicaciones.
- RIADM4. *Ciclos.* Nombre, fecha de inicio, fecha de fin, descripción.
- RIADM5. *Producto.*
- Identificación del documento de requerimientos, del documento de diseño, de implementación y de los casos de prueba.
 - Identificación para cada requerimiento, módulo de diseño e implementación, así como casos de prueba.
 - Fecha para la que deben estar listos los productos.
 - Versión.
 - Fecha.
 - Iteración en que se desarrollara.
 - Racional.
 - Descripción.

- Racional de la nueva versión.
- RIADM6. *Proceso*
 - Duración de las fases.
- RIADM7. *Cambios*.
 - Persona que identifica y que realiza el cambio, los elementos que resultan afectados (por referencias cruzadas) y la fecha o intervalo de fechas en que se registrarán los cambios sobre las entidades del desarrollo.

IV.2.1.2. Requerimientos funcionales

Los requerimientos funcionales describen lo que debe hacer el sistema [Sommerville, 2002]. A continuación se listan estos requerimientos para el sistema SARE.

IV.2.1.2.1. Formulación

- RFF1. *Configurar métricas*. El sistema provee un catálogo de métricas según los elementos de la IS. A partir de dicho catálogo se definen las métricas a utilizar durante el proyecto o bien durante un ciclo de desarrollo.
- RFF2. *Asignación de responsables*. El sistema permite la asignación de los ingenieros responsables que se encargan del seguimiento de una métrica particular. De forma predeterminada los responsables de las métricas son los ingenieros que están encargados según la entidad de la IS en donde se aplica la métrica.

IV.2.1.2.2. Recolección

- RFR1. *Establecer estimaciones*. El sistema permite la captura de un intervalo de valores para la estimación, cada uno de estos valores depende de un conjunto de suposiciones que se registran. Las estimaciones deben apoyarse a través de tres métodos:
- *Modelo*. El sistema debe permitir la captura de modelos, los parámetros asociados a los modelos y los valores de los parámetros para así formar una memoria organizacional de los modelos que se han utilizado. Los modelos del COCOMO II son útiles para hacer estimaciones.
 - *Analogía*. Estas son las estimaciones por datos históricos. El sistema permite basarse en los valores del proyecto actual o de proyectos anteriores que se presentaron para diferentes entidades de la IS, según la métrica que se desea estimar.
 - *Experiencia personal*. Si no se cuenta con ninguna de las dos anteriores, el sistema permite la captura de las estimaciones a través de las experiencias personales.
- RFR2. *Configurar estimaciones de métricas asociadas*. El sistema debe permitir configurar estimaciones para las métricas asociadas a la estimación actual de una métrica.
- RFR3. *Configurar reestimaciones (actualizar reestimaciones)*. El sistema ofrece tres mecanismos para programar una reestimación:
- Por períodos de tiempo, por ej. un día particular de cada semana, cada mes.

- establecer un día determinado según una fecha (puede ser cuando termina una fase, un ciclo o el proyecto), o bien programar un evento (por ejemplo, si la tasa de defectos de un producto esta por arriba de un límite permisible).
 - A la conclusión de un proceso dado, por ej. una fase, un ciclo o un proyecto.
- RFR4. *Calcular el valor actual.* Se registra el valor actual que se obtiene, así como la fuente (revisiones personales, inspecciones, pruebas) por la que se obtuvo dicho valor.
- RFR5. *Estimaciones individuales o grupales.* Se debe brindar soporte para hacer una estimación grupal a través del método WideBandDelphi.

IV.2.1.2.3. Análisis

- RFA1. *Gráficas predefinidas.* El sistema provee un amplio número de gráficas predefinidas según el tipo de monitoreo, así los ingenieros disponen de una gran variedad de enfoques de la repercusión que tienen las métricas sobre: (1) otras métricas, (2) los elementos de la IS y (3) a través del tiempo.
- RFA2. *Establecer valores de referencia.* El sistema establece los límites bajo y alto según el rango de valores de la estimación de la métrica. Estos límites cambian en caso de que cambien los valores de la estimación. Con estos valores de referencia se descubren las situaciones anormales.
- *Lista de excepciones.* El sistema permite verificar a través de memoria organizacional si una anomalía ya se había presentado en monitoreos anteriores
- RFA3. *Configurar revisión del monitoreo.* El sistema permite la configuración de los tipos de monitoreos que se van a revisar. Estos tipos pueden ser:
- Monitoreo del estado actual.
 - Monitoreo de puntos de revisión. Al final de cada etapa de desarrollo, ciclo o proyecto el sistema provee los progresos que se han alcanzado contra lo que se había estimado.
 - Monitoreo continuo. El sistema permite hacer este monitoreo en cualquier momento del desarrollo y utilizar los datos que se tengan disponibles.
 - Monitoreo activado por una anomalía. Cuando ocurre una anomalía (un valor actual excede un valor estimado), entonces automáticamente se notifica a los ingenieros de software responsables dicha anomalía.
 - Monitoreo basado en tiempo. Es realiza el monitoreo de acuerdo a determinados períodos de tiempo.
- RFA4. *Configurar objetivo del monitoreo.* El sistema permite la captura del objetivo del monitoreo, este puede ser de dos tipos: explorar una teoría o confirmar una relación, para ambos se registra una referencia (literaria o empírica) que se almacena como memoria organizacional. Se explora una teoría cuando hay índices empíricos o literarios de la relación entre dos métricas y se explora una relación cuando ocurre lo contrario.

- RFA5. *Configurar monitoreos de las métricas asociadas.* El sistema debe permitir configurar monitoreos para las métricas asociadas al monitoreo de una métrica en particular.
- RFA6. *Revisar monitoreos asociados.* El sistema debe brindar la opción de presentar las gráficas de los monitoreos asociados, un monitoreo es asociado cuando corresponde a alguna de las métricas relacionadas a la métrica que se está monitoreando, o bien, es una gráfica donde se presenta el comportamiento de la métrica a través de los valores actuales de proyectos anteriores (siempre y cuando los proyectos anteriores sean del mismo tipo del proyecto actual).
- RFA7. *Recomendaciones del sistema.* El sistema provee un conjunto de recomendaciones predefinidas según los elementos de la IS, esta lista puede crecer a través de experiencias personales. Este listado predefinido resulta útil cuando no se cuenta con información histórica.

IV.2.1.2.4. Interpretación

- RFI1. *Lista de circunstancias.* El sistema permite acceder a través de una memoria organizacional una lista de circunstancias predefinidas, estas pueden ser circunstancias generales o bien circunstancias particulares (asociadas a un tipo de monitoreo particular). También el sistema debe permitir agregar nuevas circunstancias. Estas circunstancias mantienen una referencia cruzada a las excepciones (causas) particulares, en cada caso, es posible acceder a información histórica del contexto (ingeniero, tipo de monitoreo) de la asociación entre las circunstancias y las excepciones. El sistema permite actualizar o eliminar las circunstancias que ya no sean válidas.
- RFI2. *Discusiones de la interpretación.* El sistema permite que diversos ingenieros discutan cual es la interpretación final que resulta más conveniente, una vez que se obtiene esta, lo que sigue es registrarla en el sistema.
- RFI3. *Recomendaciones del sistema.* El sistema provee un conjunto de recomendaciones predefinidas según los elementos de la IS, esta lista puede crecer a través de experiencias personales. Este listado predefinido resulta útil cuando no se cuenta con información histórica.

IV.2.1.2.5. Retroalimentación

- RFRET1. *Lista de acciones de Administración.* Para cada una de las circunstancias (causas) que generan las excepciones (efectos), el sistema permite acceder a una lista de acciones de administración que son apropiadas para cada circunstancia. El sistema también permite agregar y actualizar dichas acciones de administración, así como eliminar las que ya no sean válidas.
- RFRET2. *Discusiones de la retroalimentación.* Para decidir cual debe ser la retroalimentación, el sistema debe contar con mecanismos que permitan obtener esta de forma individual o grupal. Así mismo, registrar las acciones de la retroalimentación, es decir, las acciones que se ejecutaran.
- RFRET3. *Métricas a monitorear.* El sistema permite que se etiqueten las métricas que estén asociadas a los elementos de la IS donde se aplicaran las acciones de

administración. Esto ayudara a conocer si efectivamente dichos elementos han mejorado.

RFRET4. *Registro de la ruta de la retroalimentación.* El sistema permite registrar todas las acciones previas que fueron ocurriendo y así formar caminos de éxito o de fracaso.

IV.2.1.2.6. Administración

RFADM1. *Registro del proyecto y de los ciclos.* El sistema permite dar de alta proyectos y Configurar los ciclos de desarrollo requeridos.

RFADM2. *Registro de equipos, ingenieros y roles.* El sistema administra los equipos, el objetivo de dichos equipos, los ingenieros que participan y sus roles. El rol de líder debe ofrecer la capacidad para dar de alta ingenieros de software y equipos de trabajo.

RFADM3. *Registro de productos, procesos de la IS.* El sistema permite registrar o referenciar requerimientos, módulos de diseño e implementación y casos de prueba. También permite registrar las siguientes referencias cruzadas:

- Requerimientos asociados con otros requerimientos.
- Módulos de diseño asociados a los requerimientos.
- Submódulos de diseño/implementación asociados a módulos diseño/implementación respectivamente.
- Módulos de implementación asociados a módulos de diseño.
- Casos de prueba asociados a requerimientos, módulos de diseño e implementación.

Estas relaciones resultan útiles para conocer el impacto de las métricas a su contexto de desarrollo de software.

RFADM4. *Control de cambios para los elementos de la IS.* El sistema permite registrar los cambios, con estos se sabe la estabilidad de cada elemento y también se sabe que los cambios pueden afectar a otros elementos, para cada cambio se guarda una nueva versión elemento de IS.

RFADM5. *Cambiar de proyecto.* Cuando un usuario inicia una sesión, se puede seleccionar ingresar a un proyecto en particular, o bien, una vez que se ha ingresado al sistema, también se puede cambiar de proyecto.

IV.2.1.2.7. Acceso

RFACC1. *Control de acceso.* El sistema debe controlar el acceso a SARE, debe permitir que los usuarios que ingresen al sistema solo vean la información que les es permitido.

IV.2.1.3. Requerimientos colaborativos

La naturaleza del proyecto obliga que se definan los requerimientos colaborativos para apoyar el trabajo en grupo. Estos requerimientos quedan definidos a través de las siguientes propiedades: participación, interacción, coordinación, distribución, notificación de eventos, visualización y confidencialidad, que a continuación se describen:

- *Interacción.* Trata con las comunicaciones e interacciones entre entidades. Estos requerimientos influyen directamente en los modelos de comunicación y propiedades de interacción.
- *Participación.* Se refiere a los mecanismos usados para definir la forma en que los participantes interactúan con la aplicación.
- *Visualización.* Es el criterio de presentación de la información y tiene que ver con el diseño de las interfaces.
- *Notificación de eventos.* Depende de la conciencia que tiene el sistema de los usuarios, así como los usuarios de los eventos que notifica el sistema.
- *Coordinación.* Incluye los mecanismos y reglas creadas para asistir el comportamiento de los recursos.
- *Distribución.* Se refiere a la estrategia para diseñar e implementar aplicaciones. En una misma aplicación algunas funciones pueden ser centralizadas mientras otras pueden estar distribuidas.
- *Soporte.* Es la base tecnológica para apoyar a que se cumplan los requerimientos

Según esta clasificación se identificaron los requerimientos colaborativos que se describen a continuación.

IV.2.1.3.1. Interacción

- RCI1. *Localización.* El sistema permite a los ingenieros que se encuentran localizados de manera distribuida (geográficamente) que puedan trabajar con el sistema y participar en el proceso de medición. Y así lograr que su localización no sea factor de desventaja.
- RCI2. *Capacidades de los participantes.* Existen diferentes roles y cada uno de los roles tiene derechos, restricciones y obligaciones.
- RCI3. *Convocatoria a los asistentes.* El sistema a través de correos electrónicos indica la fecha de cierre y apertura en la que se convoca a los ingenieros a participar en una discusión.
- RCI4. *Identificación de la participación.* Cada comentario debe de tener un identificador que ayude a determinar quien fue el autor de dicha contribución.

IV.2.1.3.2. Participación

- RCP1. *Tamaño del grupo.* El tamaño del grupo es especificado según la naturaleza del proyecto.
- RCP2. *Inicio de sesión.* El sistema recupera la información personalizada del usuario y se tiene acceso a la información de otros usuarios que estén en el mismo equipo de trabajo.
- RCP3. *Representación de los datos.* La información de los datos se presenta de manera idéntica a todos los usuarios de un mismo equipo.
- RCP4. *Acciones y funciones.* Se almacena la información, esta expresión de ideas sirve para obtener las conclusiones.

IV.2.1.3.3. Visualización

- RCV1. *Interfaz de grupo WYSIWIS*. (del inglés What You See Is What I See: lo que ves es lo que veo) es flexible para incorporar gráficas e información histórica que le resulten útiles a distintos ingenieros o grupos de trabajo.
- RCV2. *Activación de una ventana*. Se actualiza el icono de la ventana indicando un cambio en el monitoreo, por ejemplo, el reporte de una anomalía.
- RCV3. *Calendario*. La vista de un calendario resulta útil para conocer cual es la carga de trabajo.

IV.2.1.3.4. Notificación

- RCN1. *Conocimiento de uso compartido*. Los usuarios están conscientes de la presencia de otros usuarios, de su localización así como los cambios y acciones que ellos realicen.
- RCN2. *Representación gráfica de los participantes*. Se tiene una lista de los participantes. Cuando un ingeniero de software ingresa al sistema le aparece una lista de los miembros del equipo que están en línea, los usuarios que participan en el proyecto y los roles que tienen asociados. De esta lista, también se tiene acceso a las métricas que cada uno de ellos esta asignado.
- RCN3. *Aviso de eventos*. Los usuarios pueden asignarse tareas a ellos mismos o a otros usuarios a través del sistema. Un usuario recibe notificaciones (correos) de que tiene eventos asignados. Cuando el ingeniero ingresa al sistema, le aparecen los eventos más próximos en los que esta invitado a participar, tales como: evaluación de métricas, actualización, inspecciones o juntas de retroalimentación.

IV.2.1.3.5. Coordinación

- RCC1. *Proceso de registro en la sesión*. Todos los usuarios que acceden al sistema necesitan registrarse a la sesión.
- RCC2. *Representación visual del controlador activo*. Para saber quién es el dueño de la discusión, se utiliza un título que ayude a identificarlo.
- RCC3. *Métodos de control de concurrencia*. La concurrencia se administra por medio de candados, se hace una cola donde los comentarios se van publicando según el orden en que llegaron, esto lo controla automáticamente la base de datos.

IV.2.1.3.6. Distribución

- RCD1. *Almacenamiento de datos*. Los datos se encuentran centralizados en un servidor que se encarga de llevar el control de la aplicación, el mecanismo de correos electrónicos y la base de datos.
- RCD2. *Arquitectura de implementación*. Es una arquitectura cliente-servidor; los clientes interactúan remotamente con un sitio de Internet que cuenta con los servicios descritos en RCD1 (control de la aplicación, el mecanismo de correos electrónicos y la base de datos).

IV.2.1.3.7. Soporte

RCS1. *Canales de comunicación.* El sistema esta basado en la red de Internet y utiliza todos los recursos que ofrece esta red para realizar la implementación de este sistema.

RCS2. *Uso de herramientas de desarrollo.* Para la elaboración de este sistema no se requiere del uso de herramientas especiales.

A continuación se describen los Casos de Uso (CU) los cuales dan una visión rápida y efectiva de la funcionalidad que debe cumplir el sistema. Los CU son una representación práctica de los requerimientos según el Lenguaje Unificado de Modelado.

IV.2.2. Casos de uso

Los CU en el lenguaje de modelado unificado UML son los requerimientos del sistema traducidos a un nivel mas descriptivo que facilita el entendimiento de lo que debe hacer el sistema. Los CU consisten de dos entidades: un actor y el caso de uso.

Un actor es usualmente una persona (aunque también puede ser un dispositivo) que solicita o requiere la funcionalidad del sistema, es la entidad que incentiva el accionar del sistema, en resumen un CU describe la manera en que el actor interactúa con el sistema.

Un diagrama de casos de uso representa los escenarios principales que cubren los requerimientos del sistema.

En la descripción de un CU se especifica paso a paso las acciones que se van llevando a cabo, los flujos alternos que pueden ocurrir, los actores que participan, el objetivo y las dependencias, es decir, las acciones que debieron haber ocurrido exitosamente para que el CU se cumpla.

IV.2.2.1.1. Caso de uso “hacer estimación”

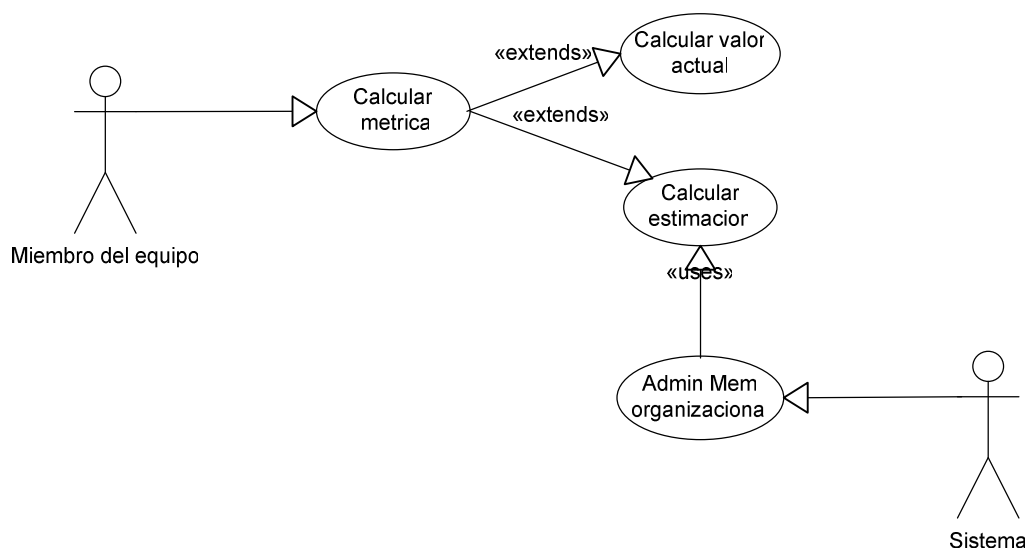


Figura 4. Diagrama del caso de uso Calcular métrica

Nombre:	<u>Hacer estimación</u>
Descripción:	Proceso para hacer una estimación de una métrica
Metas del usuario:	Calcular el valor estimado de una métrica y establecer cuando se hará una reestimación.
Actores:	<i>Usuario.</i>
Dependencias:	• Es necesario que un proyecto este dado de alta en el sistema. • Haber formulado métricas para el ciclo de desarrollo actual.

Acciones básicas:

1. El usuario accede al sistema.
2. El usuario introduce su nombre de usuario y contraseña en la interfase de acceso al sistema.
3. El usuario selecciona la opción de Recolección y el sistema presenta de forma predefinida la pantalla de recolección con las métricas de la entidad de la IS "Proyecto".
4. El usuario puede cambiar de entidad de la IS y según la que seleccione aparecen métricas asociadas.
5. El usuario selecciona la opción "Hacer nueva estimación" y un método de estimación. Según el método de estimación el sistema presenta una interfaz gráfica.
 - 5.1. El usuario selecciona estimación por analogía.
 - 5.1.1. El sistema le muestra un listado de los valores de las métricas actuales que se han presentado en ciclos anteriores del proyecto actual o bien en

proyectos anteriores.

5.1.2. El usuario selecciona uno o varios de estos datos históricos.

5.1.3. El sistema calcula el promedio de estos datos históricos y lo presenta como el valor obtenido por la estimación por analogía, como límites bajo y alto, se presenta una ventana del 50% del valor de la estimación.

5.2. El usuario selecciona estimación por modelo.

5.2.1. El sistema presenta una pantalla donde están todos los modelos que se han utilizado según la métrica de software que se quiere calcular. También el usuario puede agregar nuevos modelos y parámetros para estos modelos.

5.2.2. El usuario agrega valores para los parámetros de un modelo dado y el sistema registra la estimación.

5.3. El usuario selecciona estimación por experiencia personal.

5.3.1. El sistema le presenta al usuario una interfaz donde captura el valor estimado y una ventana de valores para la estimación según sus experiencias personales.

5.4. El usuario selecciona estimación grupal.

5.4.1. El sistema presenta la interfaz del método WideBandDelphi. En esta interfaz se especifica los participantes de dicha estimación y también para cuando esta debe estar lista, de ser necesario se pueden programar de nueva cuenta las reestimaciones. El sistema envía correos electrónicos a los participantes.

6. El sistema registra los datos de la estimación en la base de datos.

7. El sistema presenta una pantalla que permite al ingeniero configurar en que período o bajo que eventos se llevarán a cabo las reestimaciones. También es posible configurar cuales serán los monitoreos (gráficas) asociados a la métrica estimada.

8. El sistema registra en la base de datos la configuración para las reestimaciones y los monitoreos asociados.

9. El sistema presenta los eventos a través de un calendario, esto, para que el ingeniero tenga una idea clara de todos los eventos que se aproximan.

Nombre: Calcular valor actual

Descripción: Proceso para calcular el valor actual una métrica

Metas del usuario: Calcular el valor actual y verificar el impacto del valor actual.

Actores: *Usuario.*

Dependencias:

- Es necesario que el proyecto este dado de alta en el sistema.
- Haber calculado una estimación para la que se quiere obtener un

valor actual.

Acciones básicas:

1. El usuario accede al sistema.
2. El usuario introduce su nombre de usuario y contraseña en la interfase de acceso al sistema.
3. El sistema presenta un listado de estimaciones según el elemento de IS.
4. El usuario selecciona una estimación y el sistema presenta la pantalla de valor actual.
5. El usuario registra el valor actual correspondiente según el valor de la estimación de la IS.
 - 5.1. Si la estimación que se selecciona fue hecha a través de un modelo, entonces se pueden capturar los datos actuales del modelo.
6. El sistema registra en la base de datos el valor actual de la estimación.
 - 6.1. Si el valor actual difiere de la ventana de valores estimados, entonces se verifica el impacto de la métrica.
 - 6.1.1. Si hay métricas y contexto asociado a la métrica para la que se calcula el valor actual, entonces se avisa a los responsables de las métricas y se les envía un correo electrónico para que verifiquen sus estimaciones y también sus entidades de la IS.
 - 6.1.2. El sistema presenta la pantalla de la programación del monitoreo.

IV.2.2.1.2. Caso de uso “monitorear métrica”

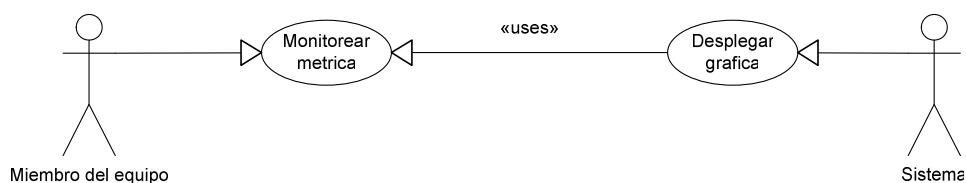
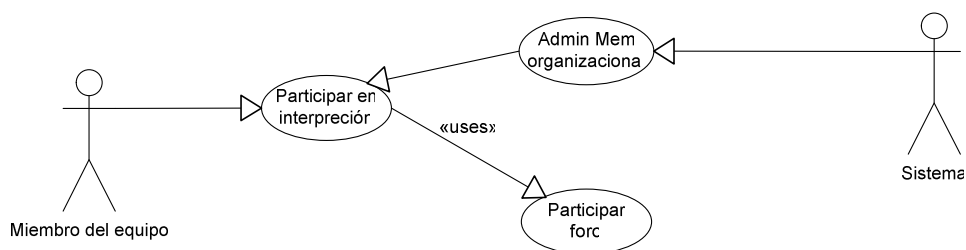


Figura 5. Diagrama del caso de uso monitorear métrica.

Nombre:	<u>Monitorear métrica.</u>
Descripción:	Proceso para visualizar el estado de las métricas que están corriendo.
Metas del usuario:	Ver el avance de una métrica.
Actores:	<i>Usuario.</i>
Dependencias:	• Es necesario que el proyecto este dado de alta en el sistema. • Las acciones de calcular valor actual han sido definidas.

Acciones básicas:

1. El usuario accede al sistema.
2. El usuario introduce su nombre de usuario y contraseña en la interfase de acceso al sistema.
3. Al entrar el usuario al sistema, el usuario puede seleccionar la opción “Monitorear métrica” para ver un listado de las métricas que están corriendo, de forma predeterminada, el sistema presenta los monitoreos de las métricas asignadas al proyecto.
4. El usuario selecciona un monitoreo y el sistema presenta una pantalla con los detalles del monitoreo.
5. El usuario descubre anomalías en el monitoreo.
 - 5.1. El usuario decide hacer un análisis personal de las anomalías y para esto revisa monitoreos asociados que le den información adicional respecto a la anomalía.
 - 5.2. El usuario decide hacer un análisis grupal y para esto convoca a una discusión grupal, una vez que se envía esta convocatoria, el sistema notifica a los participantes de la discusión.
6. El sistema permite configurar los períodos de tiempo o eventos en que se revisarán de nueva cuenta los monitoreos.

IV.2.2.1.3. Caso de uso “participar en interpretación”**Figura 6. Diagrama del caso de uso participar en interpretación.**

Nombre:	<u>Participar en interpretación</u>
Descripción:	Proceso para llevar a cabo una revisión de los valores de las métricas.
Metas del usuario:	Determinar el futuro de un producto, según sus métricas.
Actores:	<i>Usuario.</i>
Dependencias:	• Es necesario que el proyecto este dado de alta en el sistema. • Las acciones de calcular métrica están definidas. • Las acciones de monitoreo están definidas.

Acciones básicas:

1. El usuario accede al sistema.

2. El usuario introduce su nombre de usuario y contraseña en la interfase de acceso al sistema.
3. Al entrar el usuario al sistema, el usuario ingresa a una de las interpretaciones en las que esta invitado.
4. El usuario tiene acceso a interpretaciones históricas de monitoreos, interpretaciones de otros participantes y a la descripción de los valores anómalos.
5. El ingeniero a cargo de moderar las interpretaciones es quien registra las posibles causas de dichas excepciones (valores anómalos) y se toma una decisión de aceptar, rechazar o aceptar provisionalmente la entidad asociada a las métricas que están discutiendo. De existir problemas se programa una fecha para la junta de retroalimentación y se definen los participantes de dicha junta.
6. De ser necesario, el ingeniero responsable reprograma la fecha para una nueva interpretación y se registran las actividades de la inspección actual.

IV.2.3. Modelo dinámico (Diagramas de secuencia)

Durante la fase de análisis, se pueden usar los diagramas de secuencia para modelar las interacciones de los objetos que llevarán a cabo la realización de un caso de uso. Un diagrama de secuencia refleja el flujo principal de los eventos de un caso de uso.

En la fase de análisis, los diagramas de secuencia ayudan a identificar las clases que son necesarias en un sistema, así como las acciones necesarias que deben de realizar los objetos para hacer sus interacciones.

IV.2.3.1. Estimar Métrica

La Figura 7 muestra la secuencia para estimar una métrica.

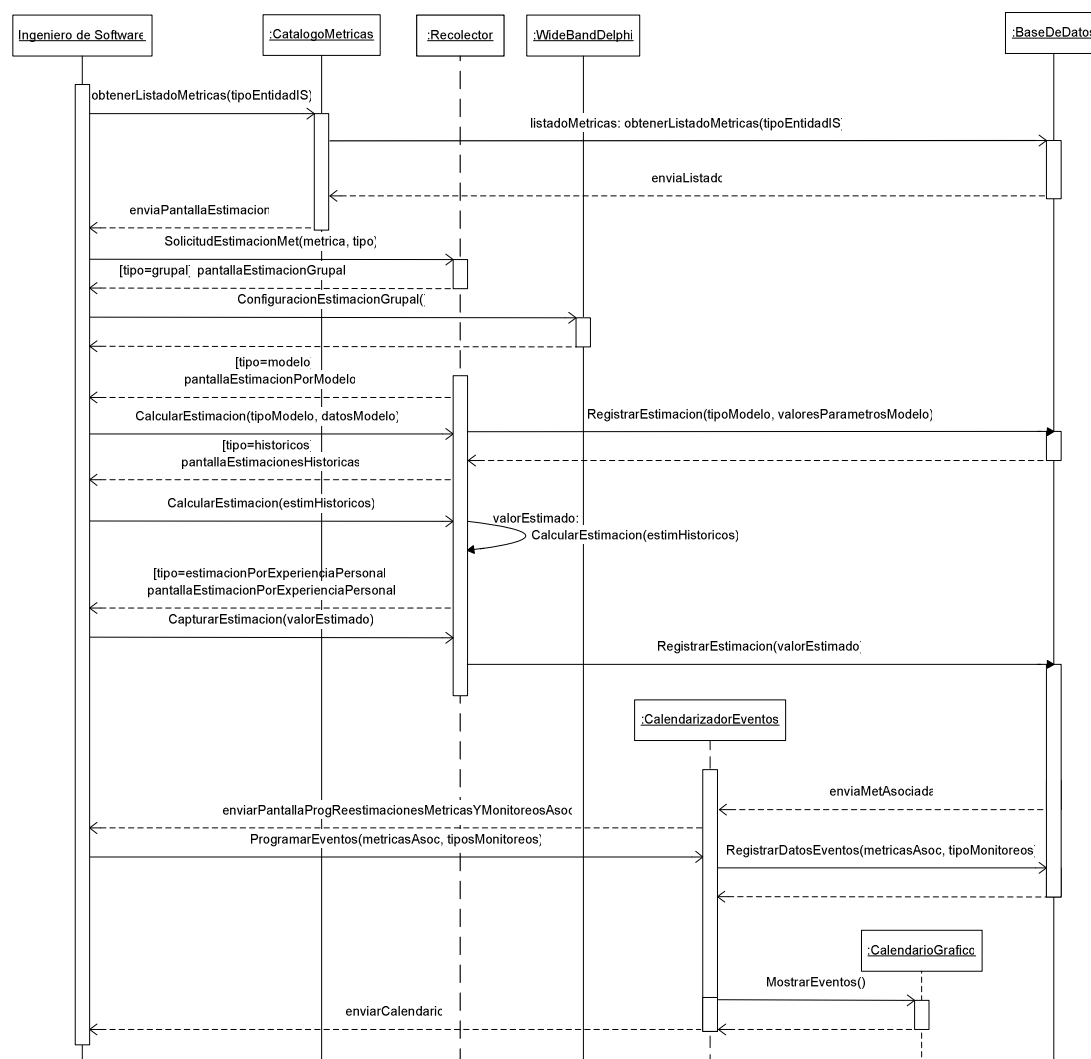


Figura 7. Diagrama de secuencia estimar Métrica.

Descripción:

1. *Ingeniero de Software* solicita el listado de métricas (según la entidad de la IS) a *CatalogoMetricas*.
2. *Catálogo de Métricas* solicita el listado de métricas a *BaseDeDatos*.
3. *BaseDeDatos* envía el listado de métricas a *CatalogoMetricas*.
4. *CatalogoMetricas* envía la pantalla de estimación a *Ingeniero de Software*.
5. *Ingeniero de Software* hace una solicitud de la estimación a *Recolector*.
6. *Recolector* filtra el tipo de estimación que llega de la solicitud.

7. Si el tipo de estimación es grupal entonces *Recolector* envía la pantalla de estimación grupal a *Ingeniero de Software*.
8. *Ingeniero de Software* configura los parámetros de la pantalla estimación grupal y solicita la configuración de la estimación grupal a *WideBandDelphi*.
9. Si el tipo de estimación es por datos históricos entonces *Recolector* envía la pantalla de estimaciones históricas a *Ingeniero de Software*.
10. *Ingeniero de Software* introduce los datos históricos para la estimación y solicita calcular la estimación a *Recolector*.
11. Si el tipo de estimación es por modelo entonces *Recolector* envía la pantalla de estimación por modelo a *Ingeniero de Software*.
12. *Ingeniero de Software* configura los parámetros del modelo de la pantalla estimación por modelo y solicita calcular la estimación a *Recolector*.
13. *Recolector* calcula el valor de la estimación.
14. Si el tipo de estimación es por experiencia personal entonces *Recolector* envía la pantalla de estimación por experiencia personal a *Ingeniero de Software*.
15. *Ingeniero de Software* introduce el valor de la estimación en la pantalla estimación por experiencia personal y solicita el registro de dicha estimación a *Recolector*.
16. *Recolector* solicita el registro del valor estimado a *BaseDeDatos*.
17. *BaseDeDatos* envía las métricas asociadas (según la estimación) a *Calendarizador de Eventos*.
18. *Calendarizador de Eventos* envía la pantalla de programar reestimaciones de métricas y monitoreos asociados a *Ingeniero de Software*.
19. *Ingeniero de Software* solicita la programación de los eventos a *Calendarizador de Eventos*.
20. *Calendarizador de Eventos* solicita registrar los datos de la programación de eventos a *BaseDeDatos*.
21. *Calendarizador de Eventos* solicita mostrar los eventos a *CalendarioGrafico*.
22. *CalendarioGrafico* envía la pantalla del calendario a *Ingeniero de Software*.

IV.2.3.2. Calcular el valor actual.

La Figura 8 muestra la secuencia para calcular valor actual.

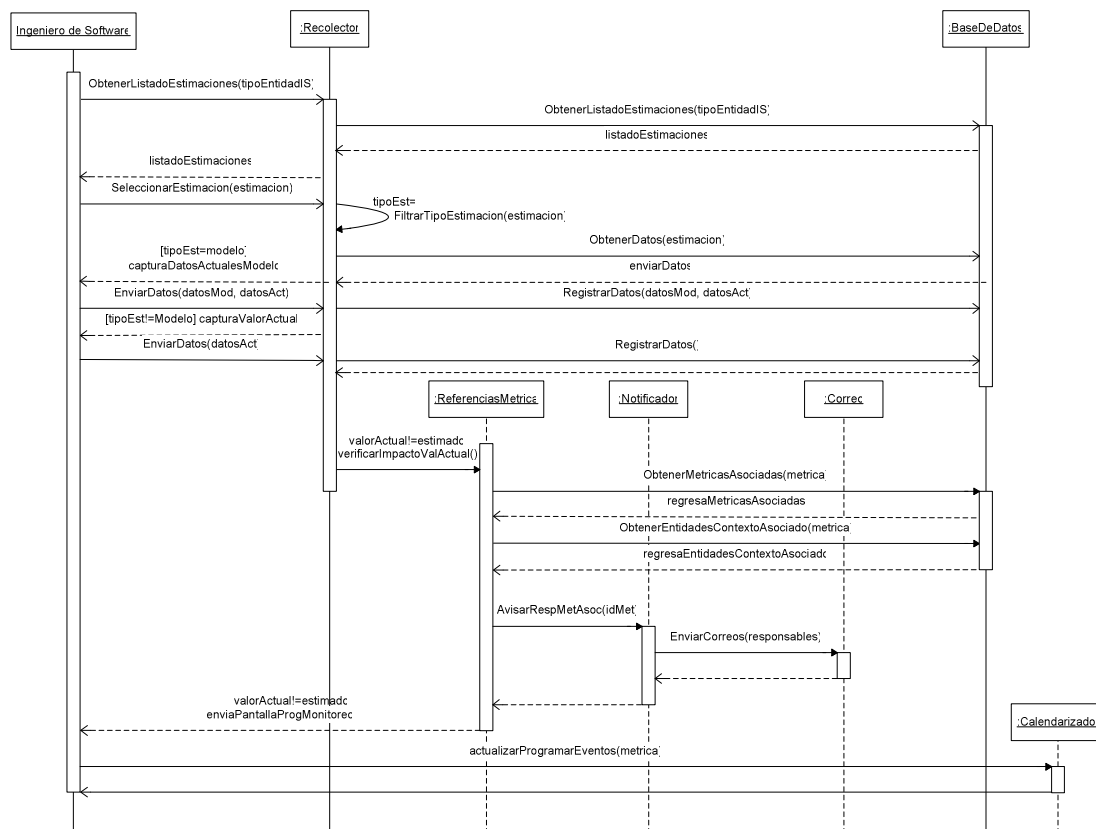


Figura 8. Diagrama de secuencia calcular valor actual.

Descripción:

1. *Ingeniero de Software* solicita el listado de estimaciones (según la entidad de la IS) a *Recolector*.
2. *Recolector* solicita el listado de estimaciones a *BaseDeDatos*.
3. *Recolector* envía la pantalla de listado de Estimaciones a *Ingeniero de Software*.
4. *Ingeniero de Software* selecciona una estimación y se la envía a *Recolector*.
5. *Recolector* filtra el tipo de estimación.
6. Si el tipo de estimación fue por modelo entonces *Recolector* solicita el modelo y los parámetros estimados a *BaseDeDatos*.
7. *BaseDeDatos* envía el modelo y los parámetros estimados a *Recolector*.
8. *Recolector* envía la pantalla de captura de datos actuales del modelo a *Ingeniero de Software*.
9. *Ingeniero de Software* envía los datos actuales de los parámetros del modelo a *Recolector*.
10. *Recolector* solicita el registro de los datos actuales de los parámetros a *BaseDeDatos*.

11. Si el tipo de estimación no fue por modelo entonces *Recolector* envía la página para la captura del valor actual a *Ingeniero de Software*.
12. *Ingeniero de Software* envía el valor actual a *Recolector*.
13. *Recolector* solicita registrar el valor actual a *BaseDeDatos*.
14. *ReferenciasMetrica* envía la pantalla de la programación de los monitoreos a *Ingeniero de Software*.
15. *Ingeniero de Software* solicita la programación de los eventos a *Calendarizador de Eventos*.
16. *Calendarizador de Eventos* solicita registrar los datos de la programación de eventos a *BaseDeDatos*.
17. *Calendarizador de Eventos* solicita mostrar los eventos a *CalendarioGrafico*.
18. *CalendarioGrafico* envía la pantalla del calendario a *Ingeniero de Software*.
19. Si el valor actual difiere del valor estimado entonces *Recolector* solicita verificar el impacto del valor actual a *ReferenciasMetrica*.
20. *ReferenciasMetrica* solicita las métricas asociadas a *BaseDeDatos*.
21. *BaseDeDatos* envía las métricas asociadas a *ReferenciasMetrica*.
22. *ReferenciasMetrica* solicita las entidades del contexto asociado a *BaseDeDatos*.
23. *BaseDeDatos* envía las entidades del contexto asociado a *ReferenciasMetrica*.
24. *ReferenciasMetricas* solicita el aviso a los responsables de las métricas a *Notificador*.
25. *Notificador* solicita el envío de correos a *ServidorCorreo*.

IV.2.3.3. Revisar monitoreo

La Figura 9 muestra la secuencia para revisar monitoreo.

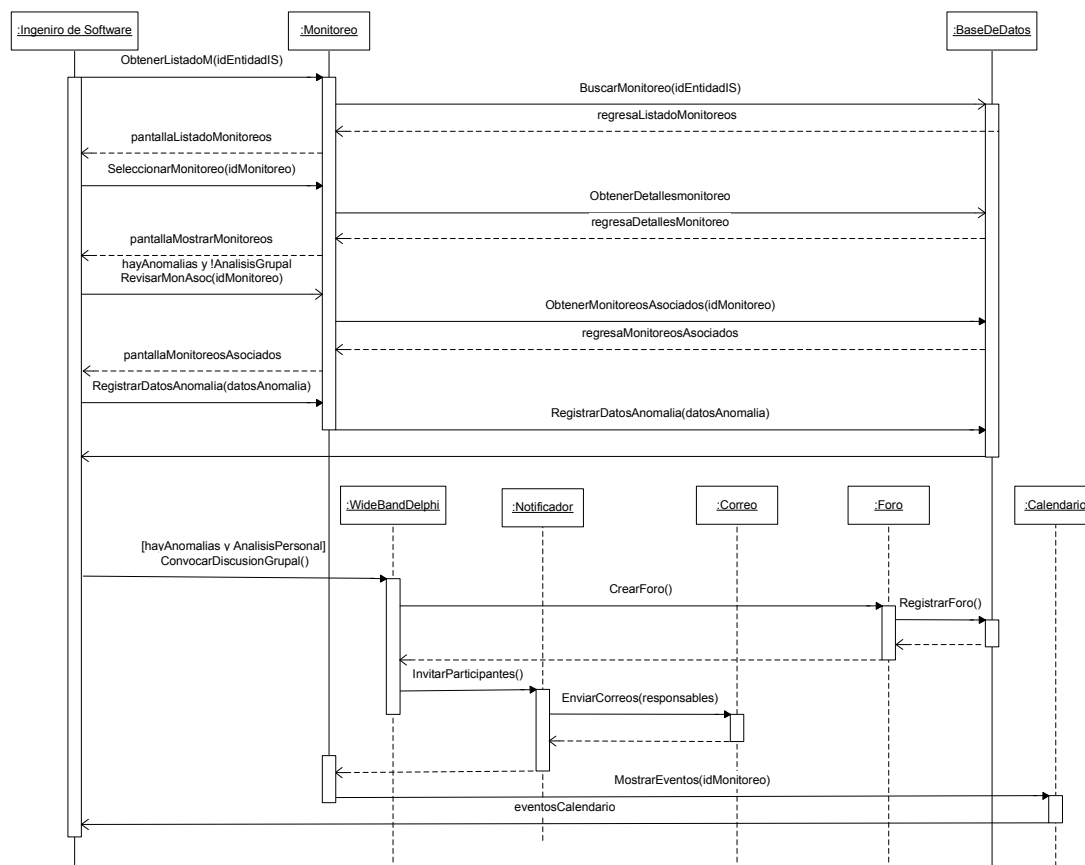


Figura 9. Diagrama de secuencia revisar monitoreo.

Descripción:

1. *Ingeniero de Software* solicita el listado de monitoreos (según la entidad de la IS) a *Monitoreo*.
2. *Monitoreo* solicita el listado de monitoreos a *BaseDeDatos*.
3. *BaseDeDatos* envía el listado de monitoreos a *Monitoreo*.
4. *Monitoreo* envía la pantalla de listado de Monitoreos a *Ingeniero de Software*.
5. *Ingeniero de Software* selecciona un monitoreo y se la envía a *Monitoreo*.
6. *Monitoreo* solicita los detalles del monitoreo a *BaseDeDatos*.
7. *BaseDeDatos* envía los detalles del monitoreo a *Monitoreo*.
8. *Monitoreo* envía la pantalla de los monitoreos.
9. Si hay anomalías y no se decide un análisis grupal entonces *Ingeniero de Software* solicita revisar los monitoreos asociados a *Monitoreo*.
10. *Monitoreo* solicita los monitoreos asociados a *BaseDeDatos*.
11. *BaseDeDatos* envía los monitoreos asociados a *Monitoreo*.

12. *Monitoreo* envía la pantalla de los monitoreos asociados a *Ingeniero de Software*.
13. *Ingenieros de Software* obtiene conclusiones de las anomalías y solicita registrar los datos de la anomalía a *Monitoreo*.
14. *Monitoreo* solicita registrar los datos de la anomalía a *BaseDeDatos*.
15. Si hay anomalías y se decide un análisis grupal entonces *Ingeniero de Software* solicita una discusión grupal a *WideBandDelphi*.
16. *WideBandDelphi* solicita crear foro a *Foro*.
17. *Foro* solicita registrar foro a *BaseDeDatos*.
18. *WideBandDelphi* solicita la invitación de los participantes a *Notificador*.
19. *Notificador* solicita el envío de correos a *ServidorCorreo*.
20. *Monitoreo* solicita mostrar los eventos a *CalendarioGrafico*.
21. *CalendarioGrafico* envía la pantalla del calendario a *Ingeniero de Software*.

IV.2.3.4. Participar en interpretación

La Figura 10 muestra la secuencia para participar en la interpretación de los monitoreos.

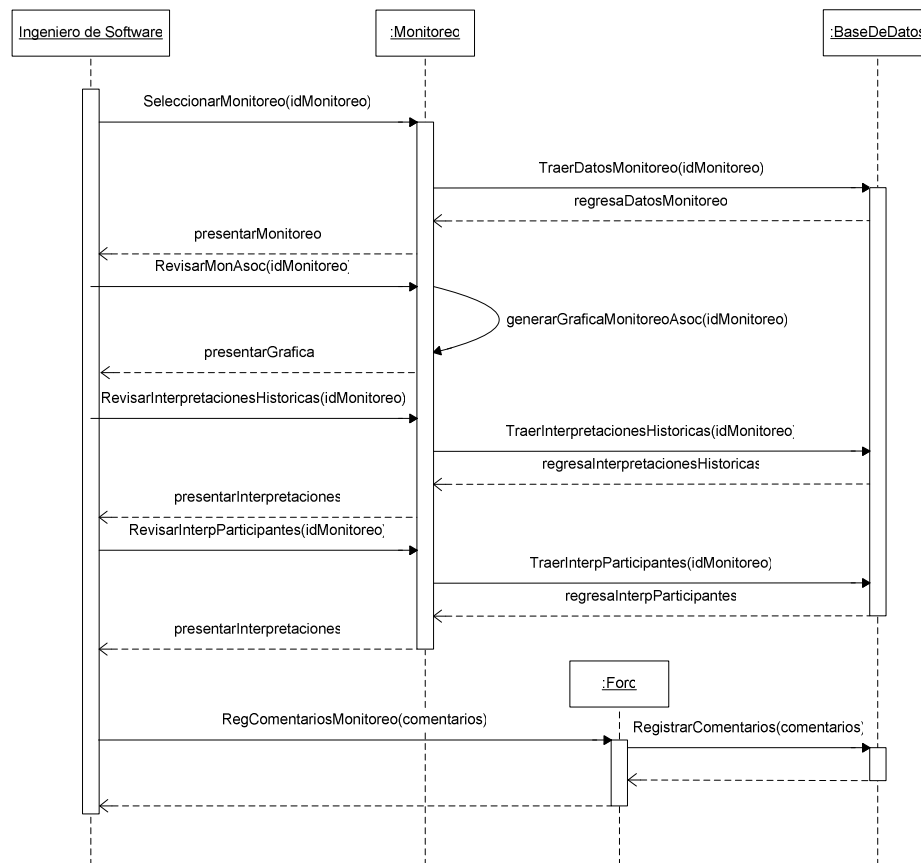


Figura 10. Diagrama de secuencia participar en Interpretación

Descripción:

1. *Ingeniero de Software* solicita un monitoreo particular a *Monitoreo*.
2. *Monitoreo* solicita los datos del monitoreo a *BaseDeDatos*.
3. *BaseDeDatos* recupera los datos del monitoreo.
4. *BaseDeDatos* envía los datos del monitoreo a *Monitoreo*.
5. *Monitoreo* envía los datos del monitoreo a *Ingeniero de Software*.
6. *Ingeniero de Software* solicita revisar los monitoreos asociados a *Monitoreo*.
7. *Monitoreo* solicita los monitoreos asociados a *BaseDeDatos*.
8. *BaseDeDatos* envía los monitoreos asociados a *Monitoreo*.
9. *Monitoreo* envía la pantalla de los monitoreos asociados a *Ingeniero de Software*.
10. *Ingeniero de Software* solicita revisar las interpretaciones históricas a *Monitoreo*.
11. *Monitoreo* solicita las interpretaciones históricas a *BaseDeDatos*.
12. *BaseDeDatos* envía las interpretaciones a *Monitoreo*.

13. *Monitoreo* presenta la pantalla de interpretaciones a *Ingeniero de Software*.
14. *Ingeniero de Software* solicita las interpretaciones de los participantes a *Monitoreo*.
15. *Monitoreo* solicita las interpretaciones de los participantes a *BaseDeDatos*.
16. *BaseDeDatos* envía las interpretaciones de los participantes a *Monitoreo*.
17. *Monitoreo* presenta la pantalla de las interpretaciones de los participantes a *Ingeniero de Software*.
18. *Ingeniero de Software* solicita el registro de los comentarios del monitoreo a *Foro*.
19. *Foro* solicita el registro del monitoreo a *BaseDeDatos*.

IV.3. Diseño

El diseño de software es el “plano”, la estrategia, el como se llevara a cabo la construcción del sistema de modo que cumpla con los requerimientos obtenidos en la fase de análisis. Durante el diseño se consideran diversas decisiones, por ejemplo:

- Acerca de la organización del sistema en subsistemas.
- Decisiones conceptuales básicas.
- Políticas.

La tarea inicial del diseño es derivar una representación de una arquitectura (una estructura global del software), [Pressman, 2002]. Esta representación sirve como marco de trabajo para actividades mas detalladas.

El diseño de subsistemas desarrolla los detalles de la arquitectura y parte el análisis en una serie de subsistemas.

A continuación se describe la arquitectura y los subsistemas del sistema SARE.

IV.3.1. Arquitectura

La arquitectura de sistemas puede definirse a diferentes niveles. Una arquitectura de nivel físico indica la forma en que la aplicación será instalada en diferentes servidores y como se llevará a cabo la comunicación entre estos. Una arquitectura a nivel lógico representa los componentes del sistema que se requieren construir. Se define la estructura y las propiedades de estos componentes así como las interacciones que mantienen entre sí [Pressman, 2002].

IV.3.1.1. Arquitectura cliente–servidor

La arquitectura cliente servidor ha proliferado, esta es la arquitectura en la que se basa Internet, tiene ventajas como que a través de clientes ligeros con un simple programa genérico (navegador). Se aprovechan características como el protocolo de comunicación TCP/IP, además del ambiente de sistema distribuido que genera Internet.

Por otro lado, se tiene una base de datos centralizada y finalmente un servidor de correo electrónico. En resumen, se trata de una aplicación web donde los clientes hacen peticiones

al servidor según datos como el rol del cliente, el proyecto en el que participa o la etapa del proceso de medición donde este capturando información (entre todas las demás acciones del cliente). El servidor responde a estas acciones y de ser necesario interactúa con la base de datos así como con el servidor de correo electrónico.

Se tienen tres servidores, el de aplicación, el de base de datos y el de correo electrónico, estos pueden estar en tres servidores físicos, sin embargo, están corriendo en una sola computadora (Figura 11).

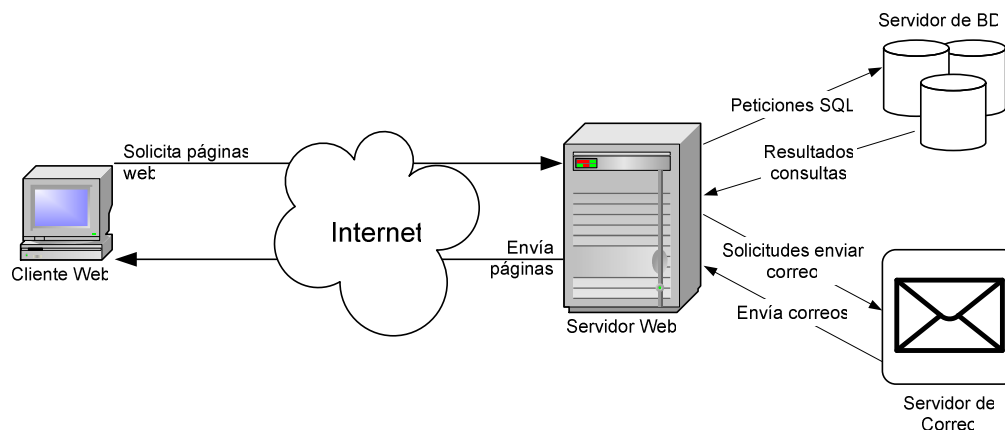


Figura 11. Arquitectura a nivel físico de SARE.

Para la arquitectura de la aplicación, se presentan como se estructura la aplicación. El patrón que se usa es el del Modelo Vista Controlador (MVC), este modelo lo que permite es poder separar la lógica de negocios de la interfaz gráfica y de las clases para la conexión a la base de datos, con esta separación se facilita el mantenimiento de la aplicación.

La capa de conexión a la base de datos, esta encapsulado por un objeto “bean” que mantiene información como las credenciales de autenticación para la base de datos. Este objeto “bean” se mantiene durante la sesión del usuario. Los objetos de tipo interfaz presentan los formularios de captura y de despliegue de información, para esto último pueden interactuar con la base de datos para la consulta de datos.

La capa controladora, básicamente dirige el flujo de las páginas, dependiendo de ciertos parámetros, se determina la página a desplegar y es la encargada de insertar o actualizar información en la base de datos. (Figura 12)

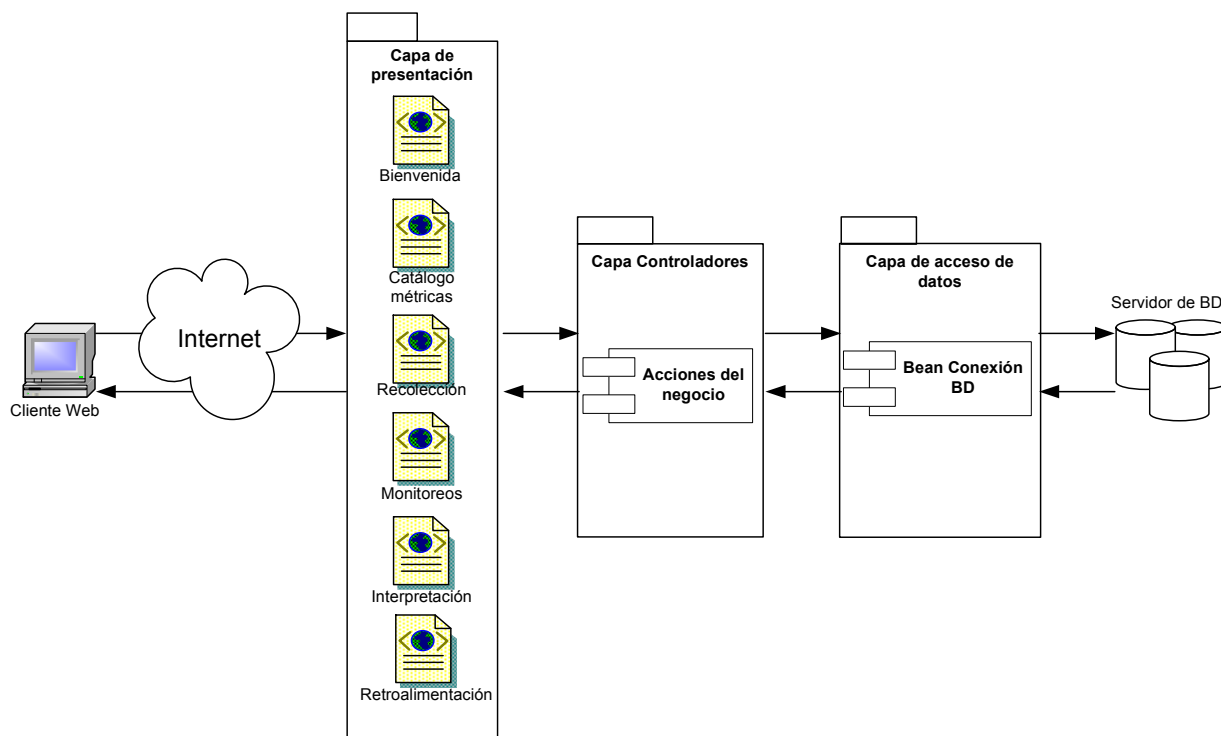


Figura 12. Arquitectura lógica y estructura de capas de SARE.

IV.3.2. Subsistemas

Los subsistemas cubren funciones asociadas a la presentación de interfaces gráficas y a la lógica de negocios que procesa los datos de entrada y de salida con que interactúan dichas interfaces. También se implementa las funciones del subsistema que se comunica con la base de datos. En SARE los subsistemas corresponden a las fases del proceso de medición y su contexto administrativo (Figura 13).

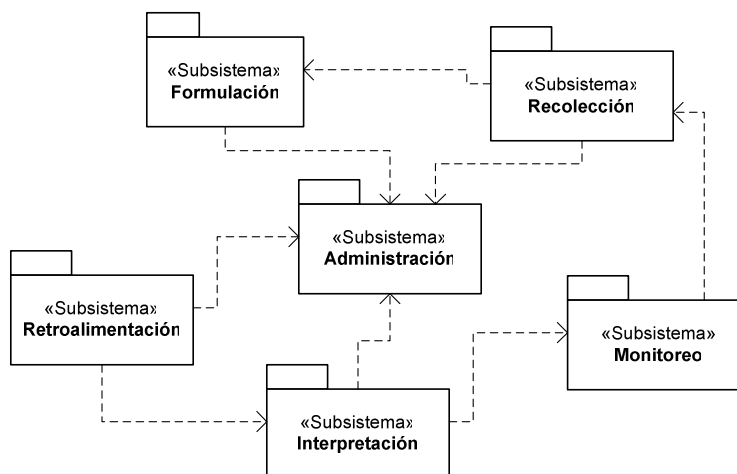


Figura 13. Diagrama de Subsistemas

IV.3.2.1. Administración de usuarios

Este subsistema se encarga de la lógica para formar y actualizar equipos de trabajo. Presenta interfaces para registrar a los ingenieros de software y asignarles propiedades tales como los roles de trabajo y equipos de trabajo a los que pertenecen.

IV.3.2.2. Administración de productos de la ingeniería de software

Este subsistema se encarga de la administración de los productos (requerimientos, módulos de diseño, de implementación y casos de prueba) de la IS, se asignan diferentes atributos para dichos productos y también se configuran las referencias cruzadas que mantienen estos productos.

IV.3.2.3. Administración de ciclos de desarrollo

Este subsistema se encarga de las funciones para generar ciclos de desarrollo, asignarlos al proyecto y establecerles una duración. Una vez creados, los ciclos pueden actualizarse o eliminarse.

IV.3.2.4. Administración del proyecto

Este subsistema se encarga de administrar los atributos del proyecto, tales como tipo de proyecto y presupuesto, entre otros.

IV.3.2.5. Control de acceso

Se encarga del acceso del ingeniero de software y las acciones que puede realizar dentro del sistema, tal como acceder a un conjunto de métricas que le fueron asignadas explícitamente o implícitamente según su rol.

IV.3.2.6. Definición de métricas

Este subsistema contiene la definición de cada una de las métricas tal como el identificador, escala, unidades de medición, métricas y monitoreos asociados, las métricas se obtienen de un catálogo que mantiene las métricas ordenadas según la entidad de la IS.

IV.3.2.7. Recolector

Este subsistema se encarga de presentar las pantallas de captura de los datos asociados al proyecto, producto, procesos, recursos, según requiera la medición que se desea obtener. Este subsistema se utiliza ya sea para hacer estimaciones o calcular el valor actual.

IV.3.2.8. Evaluador de métricas

Este subsistema es el responsable de calcular las métricas, según la definición de estas. Este subsistema se utiliza para las métricas indirectas cuyo valor se calcula según una combinación de parámetros.

IV.3.2.9. Monitoreo

Este subsistema se encarga de la generación de los datos de monitoreo, tales como: gráficas, monitoreos asociados y tipo de monitoreo, a través de estos datos, los ingenieros puedan visualizar el comportamiento de la métrica.

El sistema cuenta con gráficas predeterminadas según los elementos de la IS, a continuación se describen dichas gráficas.

IV.3.2.9.1. Producto

Para todos los productos (de todas las fases).

- Se grafican los defectos del producto que se encuentran en cada iteración. Se gráfica un análisis de residuales del valor de la predicción contra el valor actual que se obtiene.
- Gráfica del número total de productos requeridos, número total de productos listos por unidades de tiempo y número de productos en progreso por unidades de tiempo.
- Reporte del número de productos que se tiene estimado y los que se obtienen cada determinada unidad de tiempo.

IV.3.2.9.1.1. Análisis

- Se gráfica el número de solicitudes sobre los requerimientos en cada ciclo de vida de desarrollo y unidades de tiempo.

IV.3.2.9.1.2. Diseño

- Se gráfica el tamaño del diseño (número de módulos) contra el tamaño (número) de requerimientos.
- Una gráfica que relacione el número de pruebas de integración requeridas por módulo según sus métricas de estructura y el número de ligas indirectas que tiene un módulo a otros módulos como resultado de las estructuras de datos comunes.
- Una gráfica que relacione el número de pruebas requeridas de caja negra por módulo con el número de parámetros de la interfaz y el número de estados que los parámetros pueden adquirir.

IV.3.2.9.1.3. Implementación

- Gráfica del crecimiento del código fuente.
- Gráfica que relacione el número de pruebas de caja blanca por módulo con el número de ramas en un módulo y el número de acceso a datos en un módulo.
- Gráfica del tamaño del código (LOC) contra el esfuerzo de codificación (horas).
- Gráfica del tamaño del código (LOC) contra el tamaño del diseño.
- Gráfica del tamaño del código (LOC) contra el esfuerzo de codificación (horas).
- Gráfica del código que se crea (código que es nuevo) contra el código que se actualiza.
- Gráfica del código que se crea (código que es nuevo) contra el código que se borra.

- Gráfica del código que se actualiza (código que es nuevo) contra el código que se borra.

IV.3.2.9.1.4. Plan de pruebas

- Gráfica del número de casos de prueba contra el esfuerzo de pruebas.
- Gráfica del número de pruebas de aceptación que han sido efectuadas.
- Gráfica del número de pruebas de integración que han sido efectuadas.
- Gráfica del número de pruebas de sistema que han sido efectuadas.
- Gráfica del número de pruebas de caja negra que han sido efectuadas.
- Gráfica de porcentaje de requerimientos que se han probado.
- Gráfica del número de pruebas de caja blanca que se han efectuado.

IV.3.2.9.2. Proceso

- Gráfica del número de personas asociadas en cada fase y por cada iteración.
- Gráfica del perfil del esfuerzo:
 - Distribución del esfuerzo entre las fases de desarrollo.
 - Horas de esfuerzo acumuladas contra unidad de tiempo.
- Gráfica del perfil de la duración (unidad de tiempo inicial y final de cada fase).
- Gráfica del esfuerzo acumulado.
- Gráfica del retrabajo por fase y por iteración.
- Gráfica del costo total (horas, personas) de la fase sobre el tiempo (unidades de tiempo) o iteraciones.
- Gráfica de la distribución de los defectos.
- Gráfica para el análisis de defectos:
 - Gráfica de defectos por módulo
 - Gráfica de defectos por tipo.
 - Gráfica de defectos por severidad.
 - Gráfica de defectos por quien los detecto
 - Gráfica de defectos por mecanismo que lo detecto

IV.3.2.9.2.1. Pruebas

- Gráfica de los defectos encontrados y arreglados contra unidades de tiempo (efectividad del proceso de las pruebas).
- Gráfica de generación de casos de prueba contra unidades de tiempo.
- Gráfica de la generación de casos pruebas y detección de defectos contra unidades de tiempo.
- Gráfica de la duración de la revisión contra el total de defectos.
- Gráfica horas de preparación contra horas de inspección.
- Gráfica de unidades de tiempo (pueden ser semanas) asociadas a la codificación y pruebas contra el tamaño acumulado del código.
- Gráfica de las pruebas cubiertas contra las pruebas planeadas.
- Gráfica del número promedio de edad de los defectos (agrupados por tipo) no corregidos.

- Gráfica de pruebas funcionales.
- Gráfica del costo total (horas, personas) de la fase pruebas sobre el tiempo (unidades de tiempo) o iteraciones.
- Gráfica de los criterios de aceptación probados.

IV.3.2.9.3. Proyecto

- Gráfica de las estimaciones de los elementos de tamaño (líneas de código (LOC), puntos de función (PF), puntos objeto (PO)) contra el tiempo, después de cada fase y en cada iteración.
- Gráfica de la estimación y valor actual del costo, esfuerzo, recursos, y duración del proyecto.
- Gráfica de las diferencias entre las estimaciones y los valores actuales para los elementos de tamaño (LOC, PF, NO) y las estimaciones de costo, esfuerzo, recursos, y duración del proyecto, según la fuente de estimación.

IV.3.2.9.4. Recursos

Para todas las fases

- Gráfica del número de personas asociadas en cada fase y por cada iteración, así como por producto y proyecto.
- Gráfica de la experiencia de los miembros del equipo de trabajo en diferentes tipos de proyecto.
- Gráfica de la fuente de donde obtuvieron dicha experiencia.
- Gráfica de la productividad histórica que se tiene en cada fase.
- Gráfica de la productividad que se predijo contra la productividad obtenida en cada fase.
- Gráfica de las horas de trabajo y horas extras invertidas en cada fase de desarrollo, en cada iteración y en los productos de desarrollo.
- Gráfica de las horas de trabajo y retrabajo invertidas en las fases de desarrollo.

IV.3.2.9.4.1. Implementación

- Gráfica del número de componentes de software que se reutilizan.
- Gráfica del número de veces en que es reutilizado un componente de software.

IV.3.2.9.4.2. Pruebas

- Gráfica del tipo de entrenamiento que ha recibido el equipo de trabajo.
- Gráfica del número de personal que esta capacitado para llevar a cabo las pruebas.

IV.3.2.10. Inspección

Al igual que los datos crudos y las métricas en sí, se pueden calcular en varias ocasiones el mismo tipo de monitoreo (por ejemplo, la misma gráfica), esta es la razón de manejar un control de las versiones en la base de datos organizacional.

IV.3.2.11. Recomendaciones

Este subsistema brinda las recomendaciones de interpretación según los elementos de la IS, a continuación se describen dichas recomendaciones.

IV.3.2.11.1. Producto

Para todos los productos (de todas las fases).

- La contabilización de defectos sobre los productos indica los productos finales que son propensos a fallar.

IV.3.2.11.1.1. Análisis

Según las solicitudes sobre los requerimientos, esto determina cuales son los requerimientos de los usuarios que es probable que cambien.

IV.3.2.11.1.2. Diseño

- La gráfica del tamaño del diseño contra el tamaño de requerimientos, ayuda para determinar que todos los requerimientos sean diseñados y todos los diseños tengan como base a determinados requerimientos.
- Las métricas estructurales y de complejidad actúan sobre representaciones del software y ayudan a predecir a priori cuales partes del sistema de software son:
 - Menos fiables.
 - Más difíciles de probar.
 - Requieren más mantenimiento.
- La contabilización de defectos ayuda a identificar los módulos que son propensos a fallar. La clasificación de información de defectos sirve para identificar módulos particularmente peligrosos, por ejemplo, aquellos que exhiban defectos de especificación y/o diseño.
- Con la gráfica que relaciona el número de pruebas requeridas de caja negra por módulo, se identifican módulos que serán más difíciles de probar y que requieren mayor análisis para saber la causa que origina la dificultad de pruebas.
- Con la gráfica del tamaño de diseño contra el esfuerzo de diseño (horas), se identifican componentes que requirieron un gran esfuerzo de diseño respecto a su tamaño, o viceversa, estas condiciones requieren ser revisadas.

IV.3.2.11.1.3. Implementación

- La gráfica que indica la extensión del código fuente indica si se ha codificado de acuerdo a los tiempos establecidos, o bien, si se han sufrido retrasos en la codificación. Se espera que las estimaciones del tamaño del código cambian a medida que progresa el proyecto; sin embargo, se debe cerciorarse que los incrementos en las estimaciones no rebasen lo que se esta destinado a producir.
- Durante la implementación inicial, el código fuente debe crecer rápidamente, después debe tener un estado estable donde alcance poco crecimiento y no debe estar creciendo el código al final del ciclo de desarrollo.

- Con las gráficas que relacionan el número de pruebas de caja blanca por módulo con el número de ramas en un módulo y el número de acceso a datos en un módulo, se pueden descubrir líneas de código que son innecesarias.
- Con la gráfica del tamaño del código (LOC) contra el esfuerzo de codificación (horas) se determina la consistencia de la implementación respecto a la complejidad de lo que se implementa.
- Con la gráfica del tamaño del código (LOC) contra el tamaño de los comentarios (líneas) se tiene un indicador de la facilidad de entendimiento y mantenimiento de lo que se implementa.
- Con la gráfica del tamaño del código (LOC) contra el tamaño del diseño se identifican si los enunciados de diseño se implementaron.
- A través de la racional del código que cambia, se pueden conocer si las razones por las que cambia el código son o no anormales.

IV.3.2.11.1.4. Plan de pruebas

Con la gráfica del número de casos de prueba contra el esfuerzo de pruebas se obtiene una indicación del retrabajo (cuando el número de casos de prueba es considerablemente menor que el esfuerzo de pruebas).

IV.3.2.11.2. Proceso

- Con la gráfica de la distribución de los esfuerzos de cada fase se refleja que este no se concentre en la fase de implementación, ya que de ser así entonces es probable que sean muchos los cambios y que los requerimientos no hayan quedado bien definidos.
- Con la gráfica del retrabajo por fase se puede obtener si las fases subsecuentes tendrán atrasos.
- Con la gráfica del número de defectos en la fase de implementación a través del tiempo, se indica si las fases anteriores acarrear problemas tales como decisiones de diseño mal tomadas, o bien los requerimientos no fueron contemplados en su totalidad.

IV.3.2.11.3. Proyecto

- Con la gráfica de las diferencias entre las estimaciones y los valores actuales se indica la precisión de las predicciones y se da una medida del grado de confianza con la que se esta preparado para problemas futuros. Si el valor de la predicción y el actual difieren considerablemente, entonces no se tiene confianza en las predicciones.
- A través de la racional de la asignación de la fecha de entrega del proyecto, el sistema provee las repercusiones de la reasignación de fecha de entrega, tal como son los ajustes al esfuerzo requerido.

IV.3.2.11.4. Recursos

Para todas las fases.

- Con la gráfica del número de personas asociadas en cada fase y por cada iteración, así como por producto y proyecto, se lleva un control de la asignación de recursos.
- Con la gráfica de la experiencia de los miembros del equipo de trabajo en diferentes tipos de proyecto se conoce si la falta de experticia es un factor que debe considerarse en el desarrollo del proyecto.
- Con la gráfica de la fuente de donde obtuvo la experiencia se puede establecer cuales son los “entrenamientos” que han dado los mejores resultados.
- Con la gráfica de la productividad alcanzada contra la esperada se puede conocer el trabajo que falta por desarrollar.
- Con la gráfica de las horas de trabajo y horas extras invertidas en cada fase de desarrollo y en cada iteración se tiene una medida del retrabajo que se ha realizado.

IV.3.2.11.4.1. Implementación

- Con la gráfica referente al número de componentes que se reutilizan se conoce el ahorro que se tendrá en el desarrollo.

IV.3.2.11.4.2. Pruebas

- Con la gráfica del personal para la efectividad de la prueba se conoce cuales son los elementos que tienen mayor efectividad para la aplicación de pruebas.
- Con la gráfica del número de personal que esta capacitado para llevar a cabo las pruebas se conoce si la falta de experticia es un factor que debe considerarse para la estimación de la duración de la etapa de pruebas.

IV.3.2.12. Retroalimentación

Este subsistema se encarga de los cambios a productos, procesos y recursos para la siguiente iteración del proyecto de desarrollo. El sistema permite que el ingeniero ubique una bandera para aquellas métricas asociados a los cambios en el proceso y verificar si ayudan a evitar los valores anómalos de las métricas.

IV.3.2.13. Notificaciones

Este subsistema se encarga de avisar a los ingenieros cuando se aproxima algún cálculo de las métricas, monitoreos, una inspección o una retroalimentación. El subsistema permite configurar los días de anticipación para efectuar los avisos. Se envían correos electrónicos notificando que se acerca la fecha de participar en alguna notificación.

IV.3.2.14. Administrador de la base de datos

Este es el módulo encargado de interactuar con la base de datos organizacional. Sus funciones principales es abrir, cerrar conexiones y enviar consultas al servidor de base de

datos. Otra función es mantener la integridad y consistencia de los datos, manejando un control de las versiones.

IV.3.2.15. Control de la configuración

Se puede tener diferentes versiones de una estimación de un monitoreo.

IV.3.3. Diagramas de clases

Los diagramas de clases definen la estructura de los objetos y el tipo de relación que mantienen entre si. Con las extensiones para sistemas web de UML, estos diagramas se utilizan para modelar la dinámica que ocurre en las aplicaciones web, con estos diagramas se indica el flujo entre páginas que puede estar dada por formularios o hipervínculos [Conallen, 1999].

IV.3.3.1. Estimar Métrica

La Figura 14 muestra las clases que intervienen al estimar una métrica.

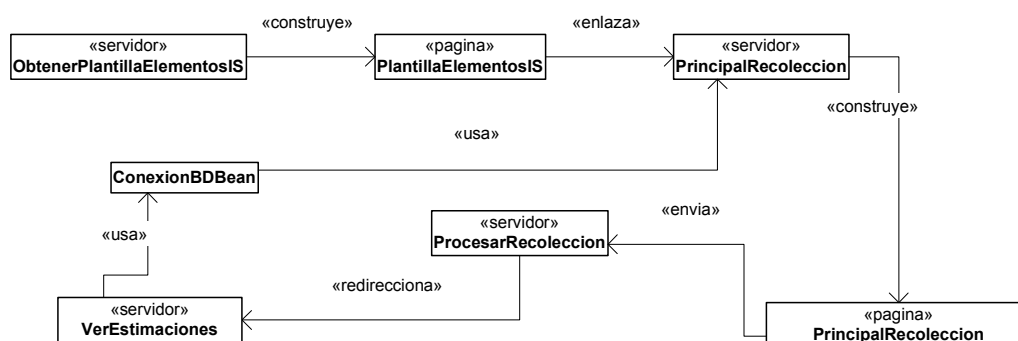


Figura 14a. Estimar métrica.

La página en el servidor *ObtenerPantallaElementoIS* se encarga de solicitar a la base de datos los elementos y subelementos de la IS para después construir la página genérica *PlantillaIS*, la cual contiene secciones para cada uno de los elementos de la ingeniería de software.

PlantillaIS permite al usuario a través de hipervínculos navegar hacia el controlador principal del flujo de control del módulo Recolección, este controlador es la página en el servidor *principalRecoleccion*, la cual a su vez construye la página en el cliente *PrincipalRecoleccion*, esta última presenta las principales opciones para la fase de recolección las cuales son: *hacerEstimacion*, *actualizarEstimacion*, y *verEstimaciones* (ver Apéndice A). En la Figura 14a. se tiene que la página *PrincipalRecoleccion* envía la opción seleccionado por el usuario (en este caso *verEstimaciones*) y en el servidor *ProcesarRecoleccion* redirecciona a la página *verEstimaciones*.

Para la acción de estimar métrica, la página *principalRecoleccion* traslada el flujo a las páginas según los métodos de estimación.

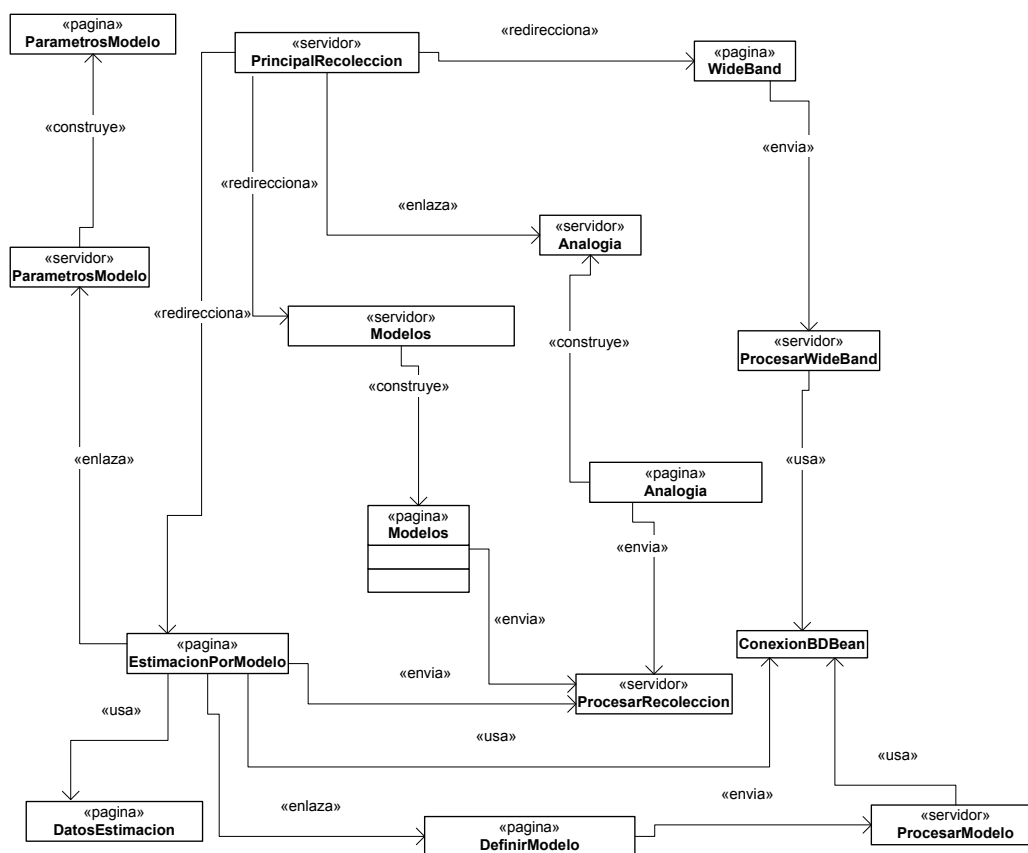


Figura 14b. Estimar métrica.

Las posibles páginas para los métodos de estimación son: *modelos* y *analogía* en el servidor y *WideBandDelphi* y *ExperienciaPersonal* como páginas sin contenido dinámico.

Los datos de la página *WideBand* se procesan hacia la página *procesarWideBand*. Las páginas de *Modelo* y *Analogía* construyen sus respectivas páginas en el cliente. Estas páginas del cliente, junto a la página de *ExperienciaPersonal*, envían sus valores a la página en el servidor *procesarRecoleccion*, esta última registra los datos haciendo uso de un componente de conexión a la BD llamado *ConexionBdBean*.

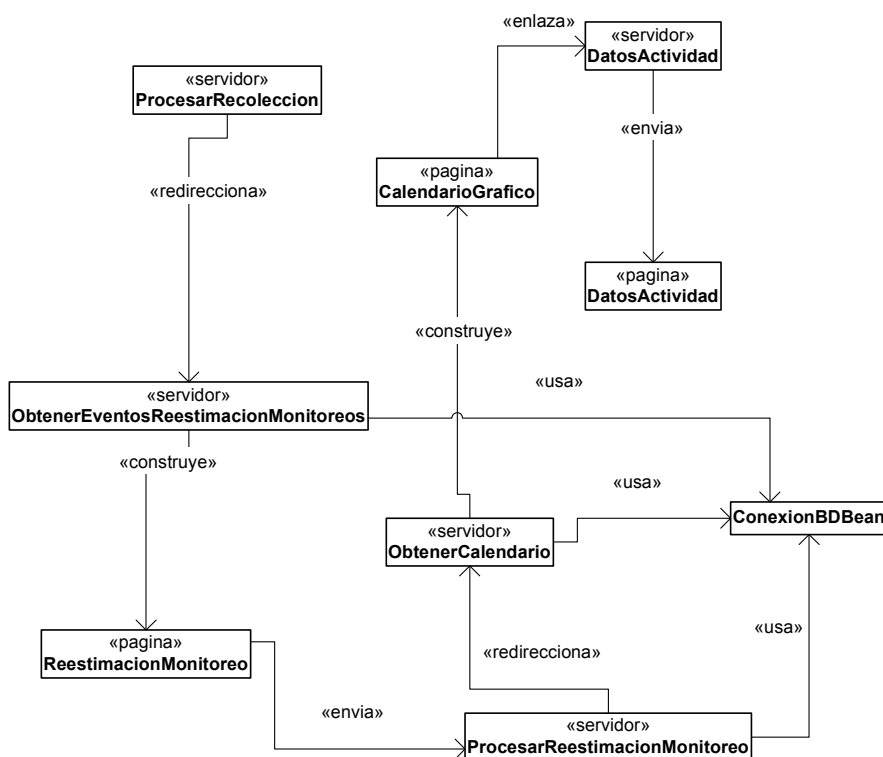


Figura 14c. Estimar métrica.

Una vez que se registran los datos, se redirecciona la página *ObtenerEventosRestimacionesMonitreo*, la cual a su vez construye la página de *reestimacionMonitoreo*, en esta página el miembro del equipo configura las reestimaciones así como los monitoreos de la métrica y de las métricas asociadas. Después de que se envía al servidor la página de *ReestimacionMonitoreo*, el servidor lo procesa en *procesarReestimacionMonitoreo*, este registra la información en la base de datos usando *ConexionBdBean*. Finalmente se registran las reestimaciones y monitoreos y se envía en el servidor a la página *ObtenerCalendario* la cual construye en el cliente una página de *Calendario* que es donde se muestran los eventos asociados a las actividades del proceso de medición.

IV.3.3.2. Calcular valor actual

La Figura 15 muestra las clases que intervienen para calcular el valor actual de la métrica.

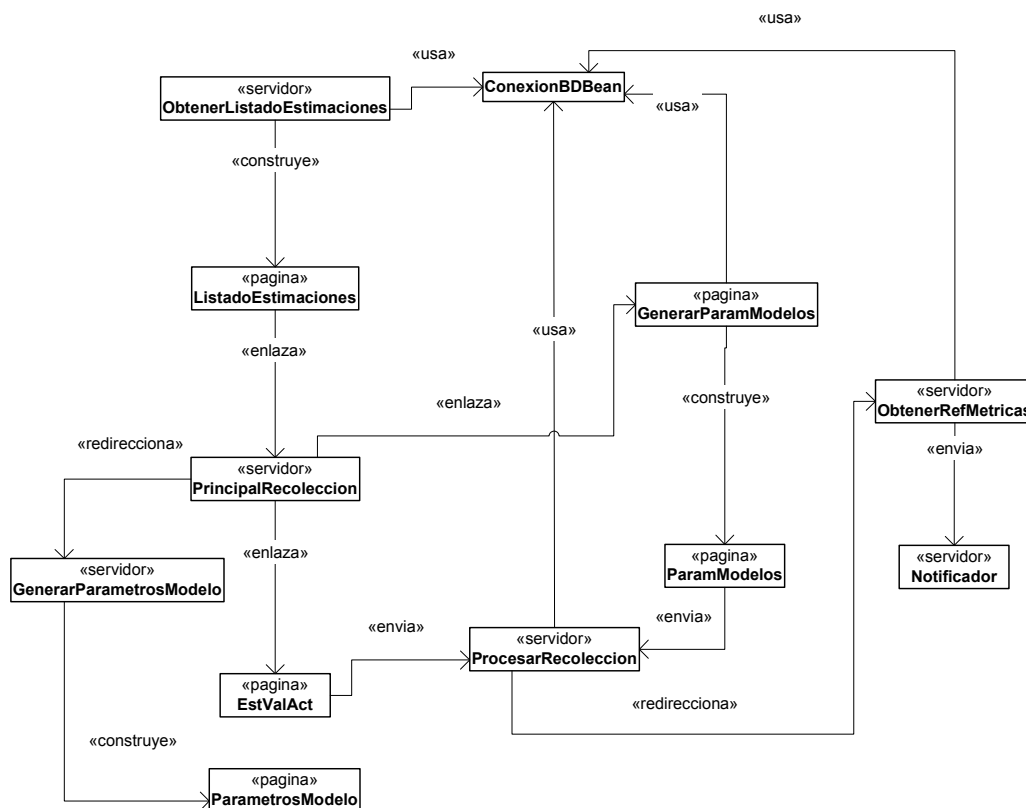


Figura 15. Calcular valor actual.

La página en el servidor *Obtener listado de estimaciones* construye la página en el cliente *listado de estimaciones*, esta página se enlaza a la página en el servidor *principalRecoleccion*, dependiendo del método con el que se hizo la estimación, se enlaza a la página de servidor *GenerarParamModelo* o a la página sin contenido dinámico *EstValAct*. La página en el servidor *GenerarParamModelo* usa a *ConexionBDBean* para obtener los valores de los parámetros del modelo, después con estos valores genera la página en el cliente *ParamModelos*.

Una vez que el ingeniero de software introduce las referencias del valor actual, ambas páginas, la de *ParamModelos* y *EstValAct* envían esta información a la página *procesarRecoleccion*, la cual es la encargada de solicitar a *ConexionBDBean* el registro del valor actual (o también valores actuales si es que se trata de un modelo). En el caso de que se presentara que el valor actual difiere de los estimados entonces se solicita al *Notificador* que avise a los responsables de las entidades y métricas que se ven afectadas.

IV.3.3.3. Revisar Monitoreo

La Figura 16 muestra las clases que intervienen al revisar monitoreo.

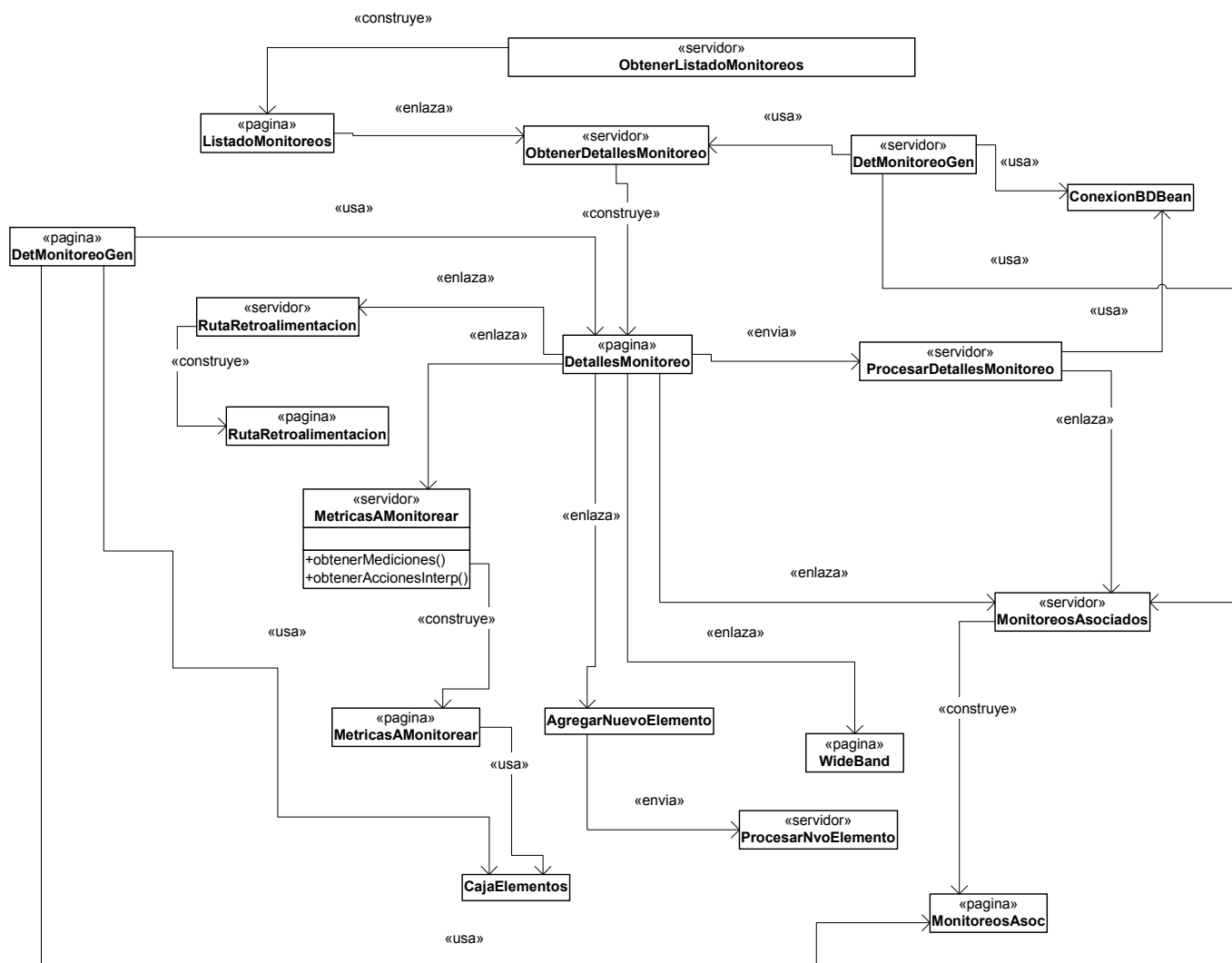


Figura 16. Revisar Monitoreo

La página en el servidor *ObtenerListadoMonitoreos* hace uso del objeto *ConexionBdBean* para construir la página *listadoMonitoreos*, a través de esta página el ingeniero selecciona un elemento del listado el cual mantiene un hipervínculo a la página de servidor *ObtenerDetallesMonitoreo*, esta usa y adquiere el comportamiento de la página en el servidor *DetMonitoreoGet* la cual hace una conexión *conexionBdBean* para mostrar los detalles del monitoreo y construir así la página en el cliente *detallesMonitoreo*. Basándose en la página genérica en cliente *DetMonitoreoGet*, *detallesMonitoreo* establece un hipervínculo con la página *monitoreosAsociados* la cual interactúa con el servidor enviando

peticiones a la página del servidor *ObtenerMonitoreosAsociados* para obtener los valores necesarios para graficar las métricas asociadas.

La página en el cliente *detallesMonitoreo* puede abrir la página de *RutaRetroalimentación*, *metricasAMonitorear*, *monitoreosAsociados*, o bien, *Agregar nuevo elemento* (donde elemento, puede ser anomalías, interpretación o circunstancias, entre otras).

Los cambios sobre los datos de *detallesMonitoreo* se envían a la página en el servidor *procesarDetallesMonitoreo*, en caso de que haya anomalías se requiera una revisión grupal entonces se envía a la página de programación de discusiones *WideBandDelphi*, una vez que se tienen los datos de la discusión se envían estos a la página en el servidor *programarWideBand*.

IV.3.3.4. Participar en interpretación

La Figura 17 muestra las clases que intervienen para participar en interpretación.

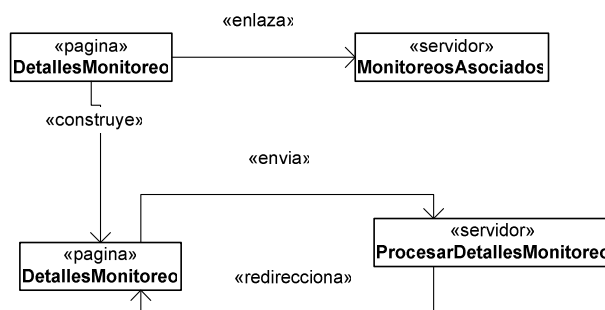


Figura 17. Participar en Interpretación.

Se registran interpretaciones en la misma pantalla donde se presentan los monitoreos, esto sirve para un contexto entre una fase y otra.

La página en el servidor *ObtenerDetallesMonitoreo* construye la página *DetallesMonitoreo* y para esto hace uso de los valores solicitados por *ConexionBdBean*. Ya que se construye la página *DetallesMonitoreo* entonces se establece la interacción con la página cliente *monitoreosAsociados* la cual solicita a la página del servidor *ObtenerMonitoreos* los monitoreos asociados. Con toda esta información el miembro del equipo es capaz de aportar comentarios sobre la interpretación del monitoreo. Una vez que el usuario envía los comentarios al servidor, este los recibe en la página *procesarDetallesMonitoreo*, la cual a su vez interactúa con la entidad *ConexionBdBean* para registrar los comentarios del miembro del equipo.

Una vez concluido el análisis y diseño, lo que sigue es presentar en el siguiente capítulo los detalles de la implementación del sistema SARE.

Capítulo V.

Implementación y funcionalidad del sistema SARE

V.1. Introducción

En este capítulo se presenta la implementación y funcionalidad del sistema SARE. Teniendo como base el capítulo anterior es posible implementar una herramienta de apoyo a la medición para equipos de trabajo distribuidos. Se describen detalles técnicos de cómo se implemento SARE y también se presenta a través de pantallas y descripciones del sistema cual es la funcionalidad obtenida con SARE.

V.2. Implementación del sistema SARE

SARE es una aplicación web cliente-servidor que se ejecuta sobre Internet y aprovecha la comunicación y el envío de información a través del protocolo TCP/IP. Para hacer uso de SARE es necesario contar con un servidor donde se instale la aplicación web y ya instalada cualquier cliente con acceso a Internet podrá usarla.

A continuación se describen las herramientas que se utilizaron para implementar SARE y también se describen las especificaciones de hardware para que SARE pueda trabajar adecuadamente.

V.2.1. Lenguaje de programación

El sistema SARE se implemento utilizando la tecnología de Java Server Pages [JSP]. Esta tecnología permite que un servidor genere de forma dinámica páginas web, JSP consiste de scripts que utilizan una variante del lenguaje java. La principal ventaja de JSP frente a otros lenguajes de scripts es que permite integrarse con clases Java (.class) lo cual permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad y dejando la parte encargada de formatear el documento html en el archivo jsp. Para los accesos al manejador de base de datos se implemento un Java bean, el cual es un componente que agrupa la funcionalidad relacionada a la comunicación con la base de datos.

En la implementación del sistema SARE se utilizaron varias librerías de software:

- CEWOLF: conjunto de clases que permiten generar graficas.
- Calendario JavaScript: Librería para usar un calendario grafico en el cliente y que permite la selección de fechas.
- Calendario JSP: Es una librería para desplegar un calendario generado dinámicamente a través de JSP.
- Foro de discusión: Son un conjunto de páginas JSP que brindan la funcionalidad para un foro de discusión.

V.2.2. Base de datos

Se utiliza el manejador de base de datos relacional MySQL [mysql] para el sistema SARE, este es el encargado de agregar, acceder y procesar datos almacenados el sistema SARE.

Se opto por utilizar este manejador de base de datos por la facilidad de uso además de ser una base de datos de código fuente abierto (gratuita) y es sencillo de manejar.

V.2.3. Servidor de correo

El sistema SARE lleva a cabo las notificaciones por medio de correos electrónicos. Se usa el paquete de java *javax.mail.** dentro de una clase de utilería en SARE para comunicar la aplicación web con el servidor de correos, este hace uso de los protocolos POP3 e IMAP para dar salida a dichos correos.

En la Tabla II se muestran las especificaciones de hardware y software para la utilización de SARE.

Tabla II. Tabla de especificación de hardware y software.

Aplicación	Hardware	Software
Servidor	• PC Pentium III 600 Mhz. o superior. • 128 Mbytes en RAM (Mínimo). • 500 Mbytes en disco duro. • Acceso a Internet.	• servidor de correo. • Servidor MySQL. • Tomcat 4.0 • JSDK 1.2.x+
Cliente	• PC Pentium II 300 Mhz. o superior. • 128 Mbytes en RAM (Mínimo). • 12 Mbytes en disco duro. • Acceso a Internet.	• Internet Explorer.

V.3. Funcionalidad del sistema SARE

Antes de empezar a describir en detalle cada una de las secciones del sistema, se presenta la estructura general de la Interfaz Gráfica del Usuario (GUI) de la aplicación.

La GUI de la aplicación SARE esta dada por una plantilla genérica (Figura 18) que divide al sistema en tres secciones, la primera sección corresponde a la parte superior y esta dada por unas pestañas, la primera de estas se refiere al contexto del desarrollo del software y tiene como etiqueta *Administración* (1), después se tienen pestañas que cubren actividades del proceso de medición (2) y por último se cuenta con la sección para la personalización de las credenciales de autenticación llamada *Mi cuenta* (3).

Después se tiene un contenedor central donde se presentan la segunda y tercera sección. La segunda sección es la parte que esta a la izquierda del contenedor central y lo que contiene

son los miembros del equipo de trabajo (4), sus roles (5) y su estado (si está o no en línea) (6).

También en esta segunda sección se presentan todos los posibles proyectos (7) y ciclos de trabajo asociados al proyecto actual para el usuario que esta en línea.

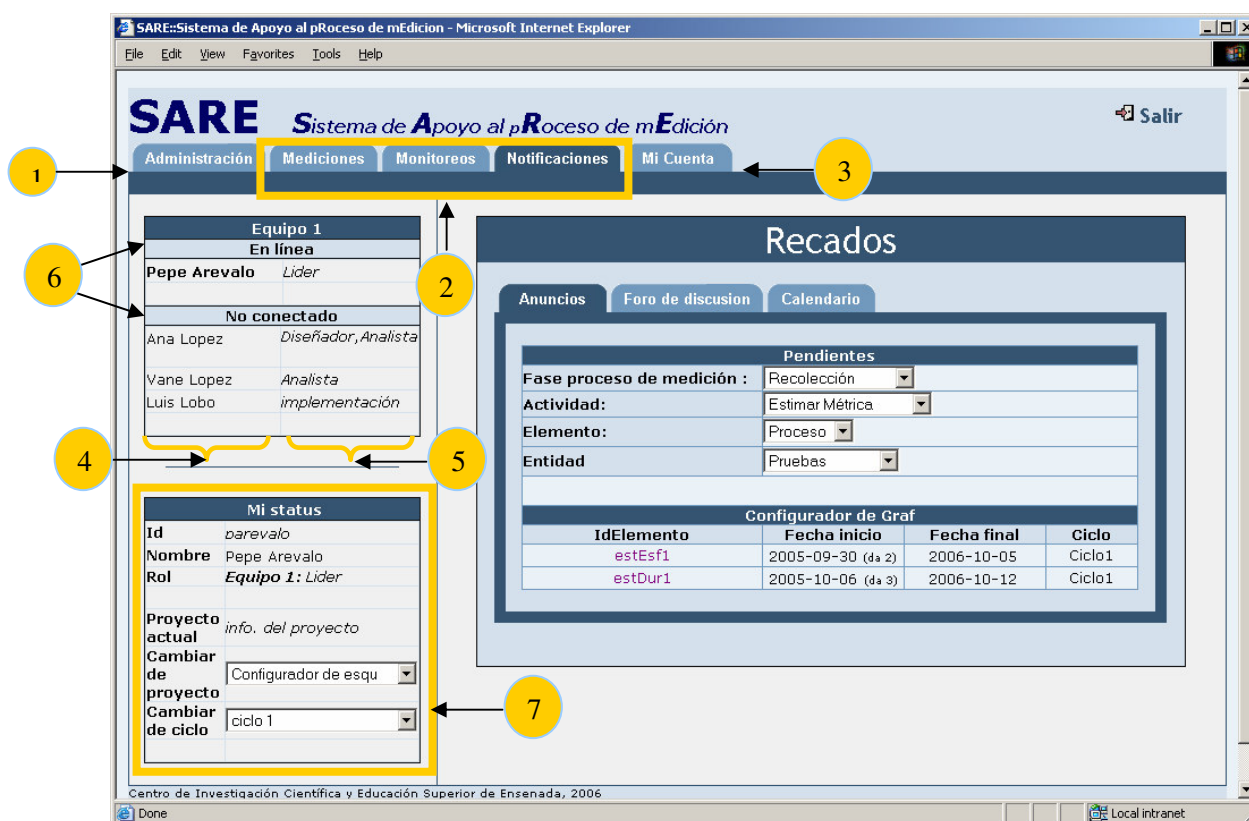


Figura 18. Plantilla genérica de SARE.

La tercera sección es la parte central del contenedor y es donde se presenta el segundo nivel de pestañas, estas están en función de la selección de la pestaña del primer nivel. Estas segundas pestañas tienen como finalidad instanciar actividades específicas de las pestañas del primer nivel.

En general, cuando se selecciona la pestaña del primer nivel aparece seleccionada de forma predeterminada la primera pestaña del segundo nivel.

A continuación se presentan las secciones del sistema, la secuencia en que se muestra es según el orden en el que se usaría el sistema.

V.3.1. Acceso al sistema SARE

Se accede al sistema a través de la pantalla de *Bienvenida* de SARE (Figura 19). Inicialmente existe una cuenta de súper usuario *administrador* (puede ser el potencial líder del equipo) el cual es el responsable de crear al menos un ingeniero de software con el rol de líder, con este rol ya se tiene la capacidad de dar de alta al equipo de trabajo y los proyectos de software. El rol del *administrador* solo esta asociado a crear equipos de trabajo y en particular a generar un primer ingeniero de software que tiene el rol de líder, de hecho, el *administrador* no esta asociado a ningún proyecto de Software en particular.

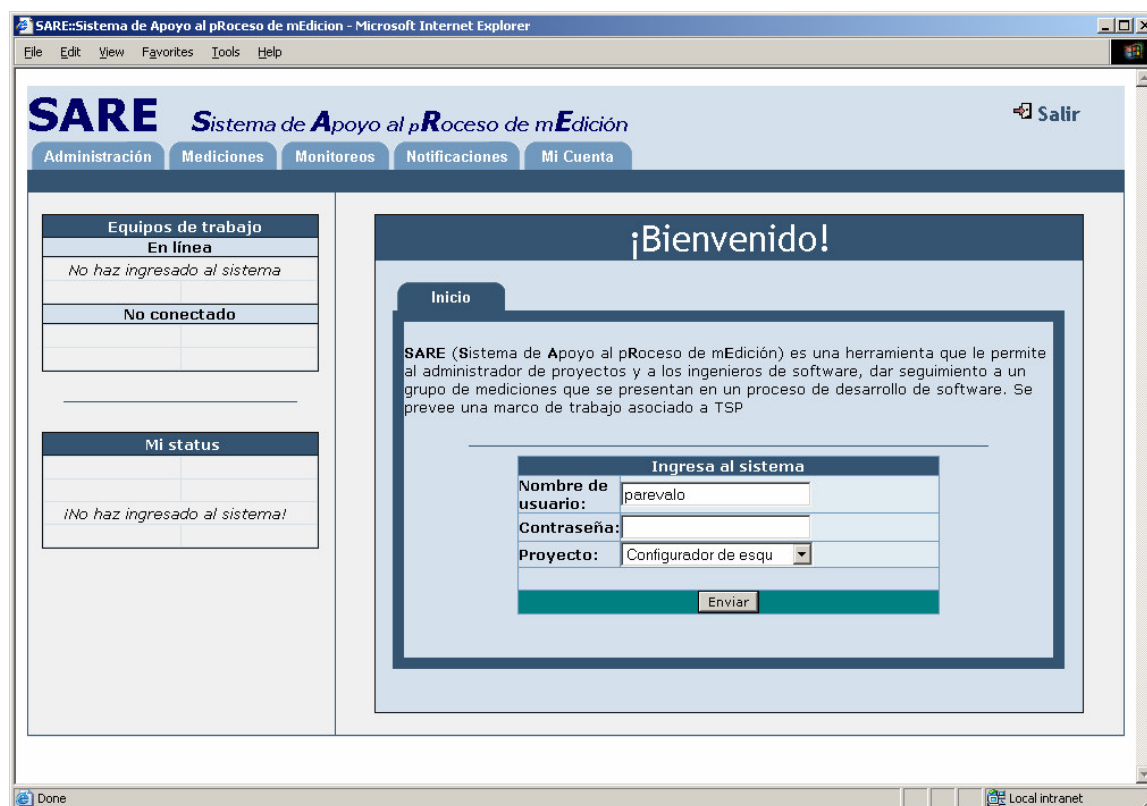


Figura 19. Pantalla de bienvenida.

Cuando el líder del proyecto ingresa por primera vez al sistema, este le envía la página para dar de alta un proyecto (en la sección 2.1 se muestran los detalles de la administración del proyecto). La otra responsabilidad del líder es referente a la creación de los equipos de trabajo (en la sección 2.2 se muestran los detalles para los equipos de trabajo).

Finalmente, un ingeniero de software ingresa al sistema según sus credenciales de autenticación y eligiendo un proyecto de la pantalla de *Bienvenida*. Inmediatamente al ingresar, el sistema le presenta la pantalla de las *Notificaciones*, es decir, los avisos de las actividades en las que el ingeniero participa, o bien es invitado a participar, esto se observa en la Figura 18.

La primera actividad que debe hacer el líder del equipo es dar de alta el contexto administrativo para las métricas de software. En la siguiente sección se describe dicho contexto.

V.3.2. Administración

Las métricas de software sirven para apoyar las actividades administrativas de un líder de proyecto o de equipos de trabajo, para auxiliar en esta tarea, SARE implementa un conjunto de subsistemas administrativos sobre los elementos del desarrollo de software, estos se describen a continuación.

V.3.2.1. Proyecto

El proyecto es una de las entidades más importantes en SARE y es la base inicial a partir de la cual se liga información tal como ciclos de trabajo o requerimientos.

Cuando se selecciona la pestaña *Administración* en el primer nivel de pestañas, se tienen dos opciones, que haya o no un proyecto en SARE, en caso de no existir un proyecto y si el ingeniero que ingresa al sistema es el de líder entonces le aparece la pantalla para dar de alta el proyecto, en caso de tener algún otro rol, simplemente el sistema le muestra un mensaje donde se indica que no se cuenta aún con un proyecto. De existir un proyecto entonces aparece de forma predefinida en el contenedor central la información del proyecto (Figura 20). Esta pantalla se divide en dos secciones, en la primera se muestran los datos del proyecto, entre estos datos los mas importantes son las restricciones de presupuesto (1), el número de ingenieros (2), la fecha de inicio (3), fecha de fin (4), tipo (5) y subtipo del proyecto (6). En la segunda sección aparecen los ciclos de trabajo (7), se puede tener un número ilimitado de ciclos, sin embargo, no deben salirse de los rangos de la fecha de inicio y fin del proyecto.

The screenshot shows the 'Admin del proyecto' interface. At the top is a navigation bar with tabs: Cfg, Usuarios, Metri asoc, Reqs, Modulos, and Pruebas. The main content area is divided into three sections: Identificación, Restricciones, and Iteraciones.

Identificación

Nombre:	Configurador de esqu		15
Tipo de proyecto:	Sistemas de negocios	Subtipo de proyecto:	Sistema de Intranet
Descripción:	Configurador de graficas		

Restricciones

Fecha de inicio:	2005-09-19	Fecha de fin:	2006-07-21
Presupuesto:	15000 p	Máximo esfuerzo:	5 personas

Iteraciones

No de iteración:	ciclo 2		
Fecha de inicio:	21/11/2005	Fecha de fin:	21/11/2005

Below the Iteraciones table, there is a list of cycles:

- ciclo 2(21/11/2005,17/02/2006)
- ciclo 1(19/09/2005,18/11/2005)
- ciclo 3(20/02/2006,21/07/2006)

At the bottom of the form are two buttons: 'Crear nuevo proyecto' and 'Actualizar'.

Numbered callouts (1-7) point to specific elements: 1 points to the 'Presupuesto' field; 2 points to the 'Máximo esfuerzo' field; 3 points to the 'Fecha de inicio' field; 4 points to the 'Fecha de fin' field; 5 points to the 'Tipo de proyecto' field; 6 points to the 'Subtipo de proyecto' field; 7 points to the list of cycles.

Figura 20. Pantalla de administración.

Por último, si el líder del proyecto es quien entro al sistema, entonces le aparecen los botones para crear un nuevo proyecto, actualizar el proyecto o bien borrarlo. Estas opciones no se presentan para ingenieros que no tengan el rol de líder.

V.3.2.2. Equipos de trabajo

Los equipos de trabajo se presentan cuando se selecciona la pestaña *Usuarios*. Se siguen las mismas políticas de roles que se siguen en el proyecto solo que ahora aplicados a equipos de trabajo.

Si hay equipos de trabajo en el sistema, al seleccionar la pestaña *Usuarios* se visualizan estos, sin embargo si no los hay y el usuario que ingresa al sistema es el líder, entonces se presenta la pantalla para dar de alta a los equipos. En la visualización de los equipos se tiene la opción para la actualización de estos. Esta pantalla se compone de tres secciones: dar de alta ingenieros de software (Figura 21), crear equipos (Figura 22), asignar equipos y roles (Figura 23).

The screenshot shows a web form titled "Crear nuevo usuario". It contains three input fields: "Nombre:" (labeled 1), "NICK:" (labeled 2), and "Correo:" (labeled 3). Below these fields is a button labeled "Agregar usuarios al equipo" (labeled 4). The form has a light blue header and a teal footer.

Figura 21. Creación de un nuevo ingeniero de software.

La alta de los ingenieros de software incluye el nombre del ingeniero (Figura 21 (1)), nick(Figura 21 (2)) y su correo electrónico (Figura 21 (3)), estos ingenieros se van agregando por medio el botón *Nuevo Ingeniero* (Figura 21 (4)) al listado de ingenieros disponibles (Figura 23 (2)).

The screenshot shows a web form titled "Crear nuevo equipo". It contains two input fields: "Nombre:" (labeled 1) and "Objetivo:" (labeled 2). The "Objetivo:" field is a text area containing the text "Un equipo cuya finalidad es". Below these fields is a button labeled "Agregar equipo al proyecto" (labeled 3). The form has a light blue header and a teal footer.

Figura 22. Creación de un nuevo equipo.

La creación de un equipo esta dada por el nombre del equipo (Figura 22 (1)) y el objetivo del equipo (Figura 22 (2)). Cada vez que se van agregando equipos a través del botón *Nuevo Equipo* (Figura 22 (3)) estos van apareciendo en el combo de los equipos (Figura 23 (1)).

Actualización de Equipos

Cfg Usuarios Metri asoc Reqs Modulos Pruebas

Actualización de equipos de trabajo

1 → Selecciona el equipo: Equipo 1 [v] [Crear equipo]

Líder (administrador) del equipo: Pepe Arevalo [v] [Crear usuarios]

2 {

Asignados al proyecto

[Empty list box]

3 {

Asignados al equipo

Ana Lopez
Vane Lopez
Luis Lobo
Pepe Arevalo

>> <<

Roles

☒ Analista ☐ Implementación

☐ Diseñador ☐ de pruebas

4 {

[Asignar participantes al equipo]

Figura 23. Asignación de ingenieros a equipos.

La asignación de los ingenieros a equipos y roles consiste en primero seleccionar algún equipo (Figura 23 (1)) y después transferir a través del botón ">>" (o bien para desasociar "<<") miembros del equipo de la lista de ingenieros disponibles (Figura 23 (2)) a la lista de ingenieros asignados al equipo (Figura 23 (3)). En cualquiera de las dos listas se puede seleccionar un ingeniero dando un clic sobre el elemento y aparecen seleccionados los roles que se han asociado a dicho ingeniero (Figura 23 (4)), se asignan los roles y estos quedan asociados al ingeniero dando un clic a otro ingeniero de cualquiera de los dos listados. A través del botón *Asignar participantes al equipo* se envían los datos a registrar al servidor y enseguida se despliega la pantalla de visualización (Figura 24).

Equipos de trabajo

Cfg
Usuarios
Metri asoc
Reqs
Modulos
Pruebas

Equipo Equipo 1			
Líder:	Pepe Arevalo		
Usuarios			
Id	Vane Lopez (<i>vane</i>)	Rol(es)	Analista
Id	Pepe Arevalo (<i>parevalo</i>)	Rol(es)	Lider
Id	Ana Lopez (<i>alopez</i>)	Rol(es)	Analista, Diseñador
Id	Luis Lobo (<i>lobo</i>)	Rol(es)	implementación
Descripción:			
	Este es el equipo 1		
Actualizar			

Figura 24. Pantalla de visualización de equipos de trabajo.

V.3.2.3. Requerimientos

Cuando se selecciona la opción de *Requerimientos* en la pestaña de segundo nivel, se presenta un listado de estos (Figura 25) así como opciones para agregar (1) y borrar requerimientos (2), para la actualización de los requerimientos se mantiene un hipervínculo en cada elemento del listado (3). Las opciones de agregar y borrar solo aparecen para cuando el sistema lo acceden el o los responsables de los requerimientos o bien el líder del equipo.

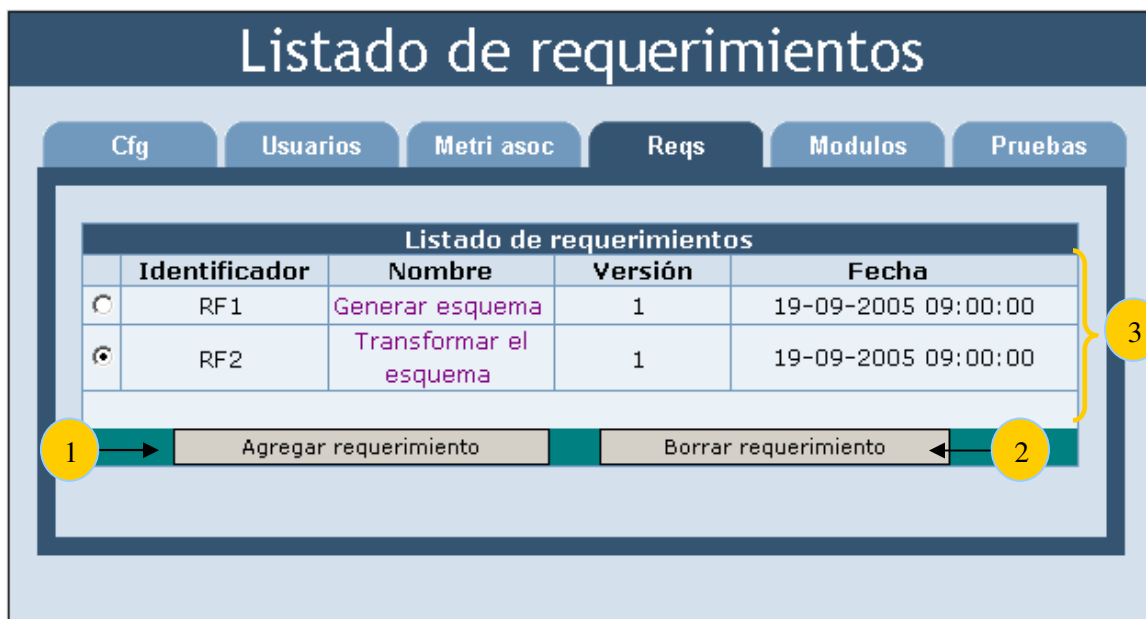


Figura 25. Listado de Requerimientos.

La pantalla de requerimientos presenta tres diferentes secciones (Figura 26), en la primera aparecen las versiones de cada requerimiento (1), se muestra en negritas la versión actual (2). En la segunda sección, se presenta la información del requerimiento (3), toda esta información se basa en los requerimientos de información de la sección IV.1.1, por último en la tercera sección esta el área donde se visualizan los requerimientos que tiene asociados.

Los botones agregar o actualizar solo aparecen para los ingenieros responsables y la GUI para la actualización o visualización es muy parecida a la alta del requerimiento.

Durante el desarrollo, los requerimientos son la guía del proyecto, sin embargo, para tener más detalle sobre el desarrollo es necesario contar con módulos a nivel de diseño y que después se convierten a módulos de implementación.

V.3.2.4. Módulos

El listado de los módulos es muy parecido a de los requerimientos, solo que dado que los módulos cubren el diseño e implementación en el listado de módulos (Figura 26) se tiene una etiqueta adicional que indica esta propiedad.

Listado de módulos				
Cfg Usuarios Metri asoc Reqs Modulos Pruebas				
Listado de modulos				
	Nombre	Tipo	Versión	Fecha
☐	Analizador datos entrada	Implementación	1	2005-09-22
☒	Definir transformacion	Implementación	1	2005-09-20
☐	Protocolo SOAP	Implementación	1	2005-09-22
		Agregar	Borrar	

Figura 26. Listado de módulos.

En la pantalla de actualización de los detalles del módulo se presentan tres tipos de asociaciones: la relación entre los módulos (Figura 27), la relación de un módulo con los submódulos (Figura 28) y las asociaciones con los requerimientos (Figura 29). Los módulos solo pueden ser administrados por los ingenieros que tienen el rol de diseñador o desarrollador según el tipo de módulo que se desea administrar.

Modulos asociados		
Modulos asociados Definir transformacion Analizador datos entrada Protocolo SOAP >> <<		
Enviar		

Figura 27. Relación entre módulos.

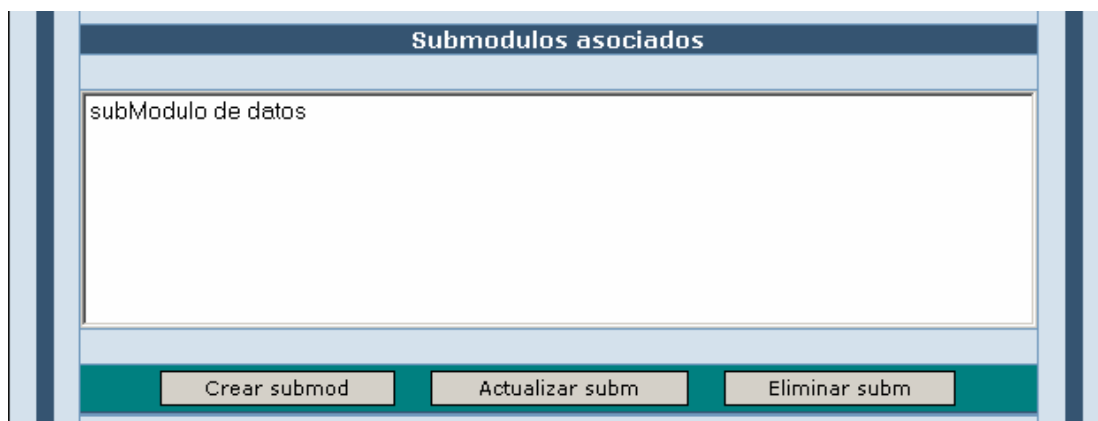


Figura 28. Relación de un módulo con submodulos.

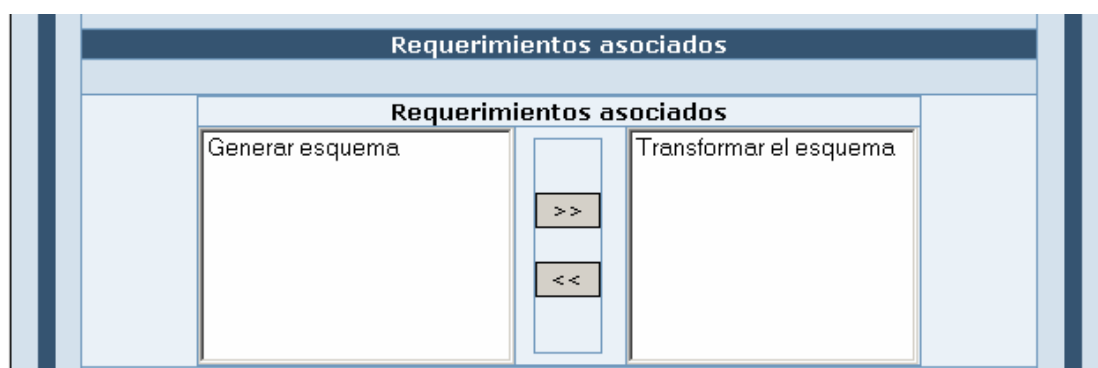


Figura 29. Requerimientos asociados.

Finalmente se tienen las pruebas del software, estas se van aplicando durante todo el proceso de desarrollo de software y la función es también administrativa, tal y como sucede con los requerimientos y los módulos. Cada caso de prueba tiene asociado un tipo de prueba según la etapa del desarrollo donde se aplica.

En la descripción del contexto administrativo se vió una constante relación entre los elementos del desarrollo. Esto ayuda a conocer el impacto de las métricas y al subsistema de notificaciones le resulta más sencillo conocer cuando es necesario avisar a los ingenieros acerca de un posible impacto de algunos valores de las métricas.

V.3.3. Formulación de mediciones

La configuración de las métricas para el ciclo de desarrollo actual se accede al seleccionar la pestaña de primer nivel *Administración* y después seleccionar la pestaña del segundo nivel *Métricas asoc.* Si ya se formularon las mediciones para el ciclo de desarrollo actual entonces se presenta la pantalla de visualización para las mediciones (Figura 30). Si el ingeniero que ingresa al sistema es el líder del proyecto entonces se presenta el botón de actualizar mediciones (1).

Métricas del ciclo

Cfg
Usuarios
Metri asoc
Reqs
Modulos
Pruebas

Proyecto	
Tam, Productividad, Duracion	
Proceso	
Análisis	Tasa de defectos, Duracion
Diseño	Tasa de defectos
Implementación	Tasa de defectos
Pruebas	Duracion, Efectividad, Tasa de defectos
Producto	
Requerimientos	Efectividad
Diseño	Efectividad
Implementación	Efectividad
Plan de pruebas	Efectividad, Tasa de defectos, Esfuerzo
Recursos	
Experiencia, Horas invertidas	
1	Actualizar mediciones

Figura 30. Métricas del ciclo.

Si no se han definido métricas para el ciclo de desarrollo actual entonces se presenta la pantalla para la alta de métricas (Figura 31). Esta pantalla consiste de dos listas desplegables, en una lista se presentan las entidades de la IS y otra es donde se muestran las subentidades asociadas (en caso del producto, proceso). También en esta pantalla se presentan las métricas disponibles (1) según la selección de las listas, estas métricas se asignan al ciclo actual usando el botón ">>" y se desasignan con el botón "<<".

Figura 31. Pantalla para el alta de una métrica.

Se tiene la posibilidad de generalizar la asignación de las métricas para los subelementos a través de la selección de la caja de tipo lista (2), un ejemplo sería para la entidad de la IS *Producto*, cuando se desea aplicar la métrica de *tamaño* a todos los productos (requerimientos, módulos de diseño, de implementación, casos de prueba). Por último las métricas tienen una vigencia asociada y esta opción se presenta por ciclo o por proyecto (3).

V.3.4. Recolección

La pantalla principal de la fase de Recolección se accede al seleccionar la pestaña de primer nivel *Mediciones* (Figura 32). La pantalla que se presenta tiene como segundo nivel de pestañas los elementos de la IS (1) así como las opciones del menú principal para las acciones de recolección, estas opciones son: estimar métrica (2), ver medición (3), calcular valor actual (4) y hacer estimación grupal (5), también aparecen las métricas (6) según la selección de la pestaña de segundo nivel (cuando se selecciona la pestaña de primer nivel

aparece de forma predefinida la primera pestaña del segundo nivel la cual corresponde a la entidad de la IS *Proyecto*). Con la selección de una métrica y de alguna opción del menú principal entonces se realiza una acción de recolección en particular, a continuación se describen las dos principales.

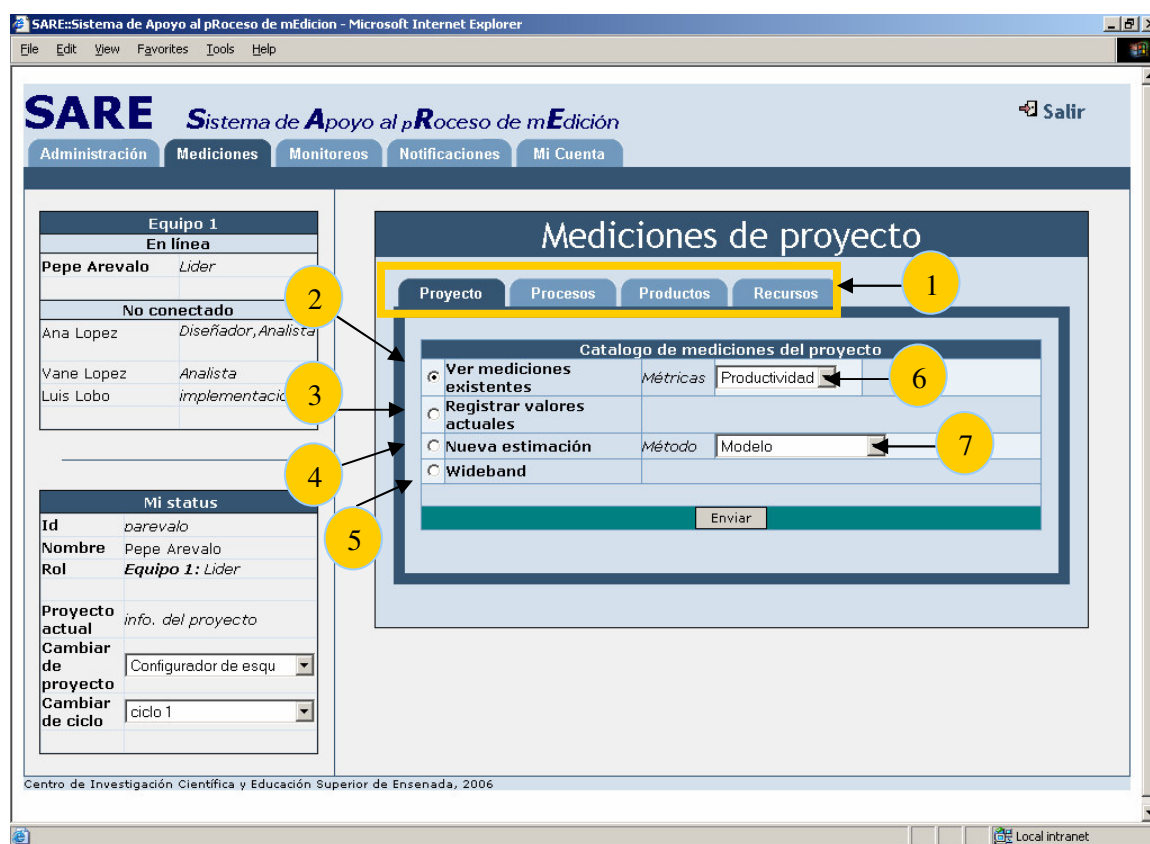


Figura 32. Pantalla principal de mediciones (recolección).

V.3.4.1. Estimación

Los métodos de estimación se presentan en un listado (Figura 32 (7)), si se selecciona el método por analogía entonces se presenta un listado con todas las anomalías históricas (Figura 33), de esta interfaz se pueden seleccionar elementos del listado (1) y también se pueden ver los detalles de cada estimación, al darle clic sobre cada elemento se abre una ventana con los datos históricos (Figura 34). Una vez seleccionados los elementos se presiona el botón de *enviar* (Figura 33 (2)) y el servidor calcula el valor estimado.

SARE: Sistema de Apoyo al pRoceso de mEdición - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Equipo 1
En línea

Pepe Arevalo	Lider
--------------	-------

No conectado

Ana Lopez	Diseñador, Analista
Vane Lopez	Analista
Luis Lobo	implementación

Mi status

Id	parevalo
Nombre	Pepe Arevalo
Rol	Equipo 1: Lider
Proyecto actual	info. del proyecto
Cambiar de	Configurador de esqu
Cambiar de ciclo	ciclo 1

Datos históricos

Proyecto Procesos Productos Recursos

Identificador	Valor	Ciclo	Fecha
☐ VA_dur_PruebasCiclo1	12 m	ciclo 1	2005-11-18
☐ VA_dur_PruebasCiclo2	15 m	ciclo 2	2006-02-17
☐ VA_dur_PruebasCiclo3	13 m	ciclo 3	2006-07-21

Identificador	Valor	Ciclo	Fecha
☐ VA_dur_PruebasCiclo1	16 m	ciclo 1	2004-01-05
☐ VA_dur_PruebasCiclo2	17 m	ciclo 2	2004-05-03
☐ VA_dur_PruebasCiclo3	14 m	ciclo 3	2004-09-01
☐ VA_dur_PruebasCiclo4	11 m	ciclo 4	2004-12-01

Identificador	Valor	Ciclo	Fecha
☐ VA_dur_PruebasCiclo1	21 m	ciclo 1	2002-04-01
☐ VA_dur_PruebasCiclo2	8 m	ciclo 2	2002-07-02

Identificador	Valor	Ciclo	Fecha
☐ VA_dur_PruebasCiclo1	17 m	ciclo 1	2001-06-01
☐ VA_dur_PruebasCiclo2	9 m	ciclo 2	2001-09-03
☐ VA_dur_PruebasCiclo3	14 m	ciclo 3	2001-12-03

Enviar

Figura 33. Listado de anomalías históricas.

Valor de la Reestimación/Valor Actual	
Proyecto	Configurador de esqu
Ciclo	ciclo 1
Historia est/val act	1
Valores	min: 5 max: 15
Met. Estimación	Analogía
Históricos	
Id valor actual	Nombre Proyecto
VA_dur_PruebasCiclo2	Configurador de Graf
VA_dur_PruebasCiclo4	Generacion Graficas
Racional	Disminucion de recursos
Cerrar	

Figura 34. Detalle de datos históricos.

Después se presenta la pantalla donde se programan las reestimaciones de las métricas (Figura 35) y los monitoreos asociados (Figura 36), para las reestimaciones de las métricas se tienen cajas de selección para periodos de tiempo (Figura 35 (1)) o bien para eventos

(Figura 35 (2)) y para los monitoreos asociados se programa el tipo de monitoreo, el cual puede ser relación entre métricas, monitoreos a través del tiempo (Figura 36 (1)) y monitoreo por componentes (Figura 36 (1)).

Mediciones del proyecto

Proyecto
Procesos
Productos
Recursos

Configuración de la reestimación

Reestimación: 1	☒ fin de ciclo	(días)	Eventos: ☒ cambio reestimacion	
	☒ fin de proyecto		☒ cambio valor actual	
	☒ fin proceso		☒ cambio de entidad	
	☒ cada semana			
	☒ cada mes			
Monitoreos: (Tam)	☒ Tam vs Productividad		☒ Tam vs Duracion	
	☒ por Tiempo		☒ por Componentes	
Revisiones:	☒ fin de ciclo	(días)		
	☒ fin de proyecto			
	☒ fin proceso			
	☒ cada semana			
	☒ cada mes			

Figura 35. Reestimaciones de las métricas.

Métricas asociadas			
Métrica:	Productividad ▾		
Reestimación:	☒ fin de ciclo (días)	Eventos:	☒ cambio reestimacion
	☒ fin de proyecto		☒ cambio valor actual
	☒ fin proceso		☒ cambio de entidad
	☒ cada semana		
	☒ cada mes		
Monitoreos:	☒ por Tiempo	☒ por Componentes	
Revisiones:	☒ fin de ciclo (días)		
	☒ fin de proyecto		
	☒ fin proceso		
	☒ cada semana		
	☒ cada mes		
Enviar			

Figura 36. Monitoreos de las métricas asociadas.

Una vez que el ingeniero de software programa los eventos correspondientes y envía los datos al servidor para su registro entonces el sistema presenta una vista del calendario (Figura 37) con todas las actividades que están programadas.

Calendario

Anuncios

Foro de discusion

Calendario

Agosto 2006





Domingo	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
		1	2	3	4	5
6	7 r	8	9	10 va	11 m	12
13	14 r	15	16	17	18	19
20	21 r	22	23	24 va	25 m	26
27	28 r	29	30 r va	31		

Figura 37. Calendario de actividades.

Los ingenieros responsables de las métricas son los que tienen autorizado hacer estimaciones sobre las métricas correspondientes.

V.3.4.2. Calcular valor actual

El ingeniero accede a la pantalla principal de recolección de la misma forma que lo hizo para la Estimación (V.3.4.1). Una vez que se selecciona una métrica en particular y una subentidad de IS (en caso de aplicar) y se selecciona la caja de calcular valor actual entonces el sistema le muestra el listado de estimaciones según lo seleccionado (Figura 38).

Listado de estimaciones			
Id	valorest	versión	fecha
est_durc1	5	2	2005-11-01
est_durs1	10	3	2005-12-12

Enviar

Figura 38. Listado de estimaciones.

El ingeniero selecciona una estimación de la lista y según el método con que se hizo la estimación entonces son los valores que es necesario capturar, de hecho, solo cuando la estimación se realizó por un modelo es cuando se tienen que capturar los valores actuales de los parámetros del modelo, en otro caso, se presenta una pantalla (Figura 39) que contiene un campo para calcular el valor actual (1) y una caja de texto para introducir algún comentario (2).

Calculo del valor actual

Valor actual:

Comentario:

Enviar

Figura 39. Registro del valor actual.

Una vez que el ingeniero introduce el valor actual y algún comentario entonces a través del botón *Envia* se envían los datos al servidor que es donde se lleva a cabo la lógica necesaria para saber el impacto del valor actual.

Si resulta que este valor no esta en el rango de lo estimado, entonces es necesario verificar el impacto que este tiene sobre los monitoreos y en el contexto de desarrollo de software, para esto, el sistema avisa a través de correo electrónico a los responsables de las métricas asociadas y del contexto administrativo. Enseguida el sistema le envía al ingeniero (él que calculó el valor actual) la pantalla para programar los monitoreos (Figura 40), ya que se programan estos se envían al servidor y este presenta (al igual que para la Estimación (V.3.4.1)) una pantalla de un calendario con todos los eventos programados (Figura 37).

Calcular valor actual:	☒ fin de ciclo (días)	Eventos:	☒ cambio reestimacion
	☒ fin de proyecto		☒ cambio valor actual
	☒ fin proceso		☒ cambio de entidad
	☒ cada semana		
	☒ cada mes		

Figura 40. Calcular valor actual.

V.3.5. Revisar monitoreos

El ingeniero selecciona la pestaña del primer nivel *Monitoreos* y el sistema presenta en el contenedor central la pantalla del listado de los monitoreos asociados al proyecto (Figura 41), se pueden ver otros listados de los monitoreos cambiando la entidad de la IS, es decir, cambiando la pestaña del segundo nivel. Este listado se ordena por métricas y por el tipo de monitoreo.

SARE Sistema de Apoyo al pRoceso de mEdición

Administración Mediciones Monitoreos Notificaciones Mi Cuenta

Equipo 1
En línea
Pepe Arevalo *Lider*

No conectado
Ana Lopez *Analista, Diseñador*
Vane Lopez *Analista*
Luis Lobo *implementación*

Mi status
Id: parevalo
Nombre: Pepe Arevalo
Rol: Equipo 1: Lider
Proyecto actual: info. del proyecto
Cambiar de: Configurador de esqu
Cambiar de proyecto: ciclo 1

Monitoreos del ciclo

Proyecto Procesos Productos Recursos

Listado de monitoreos

Monitoreo	Fecha	Objetivo
Duracion vs Costos	2005-09-20	Confirmar Teoria
Tam vs Duracion	2005-12-30	Confirmar Teoria
Tam vs Productividad	2006-01-01	Confirmar Teoria
Duracion por Tiempo	2005-09-20	Confirmar Teoria
Tam por Tiempo	2006-01-02	Confirmar Teoria

Centro de Investigación Científica y Educación Superior de Ensenada, 2006

Figura 41. Listado de Monitoreos.

Cuando el ingeniero selecciona un monitoreo en particular entonces el sistema presenta la pantalla de los detalles del monitoreo, esta se compone de cinco secciones: gráfica y datos del monitoreo (Figura 42), anomalías (Figura 44), interpretación (Figura 46) y retroalimentación (Figura 420).

En la primera sección se presenta la gráfica (Figura 41 (1)), los posibles tipos de gráficas (Figura 41 (2)), los valores estimados y los valores actuales. Dentro de los datos del monitoreo se presenta información tal como: nombre (Figura 41 (3)), objetivo del experimento (Figura 41 (4)), comentario (Figura 41 (5)) y un botón para la sección de los monitoreos asociados (Figura 41 (6)), al dar clic al botón se abre una ventana donde se puede ver los monitoreos asociados (Figura 44).

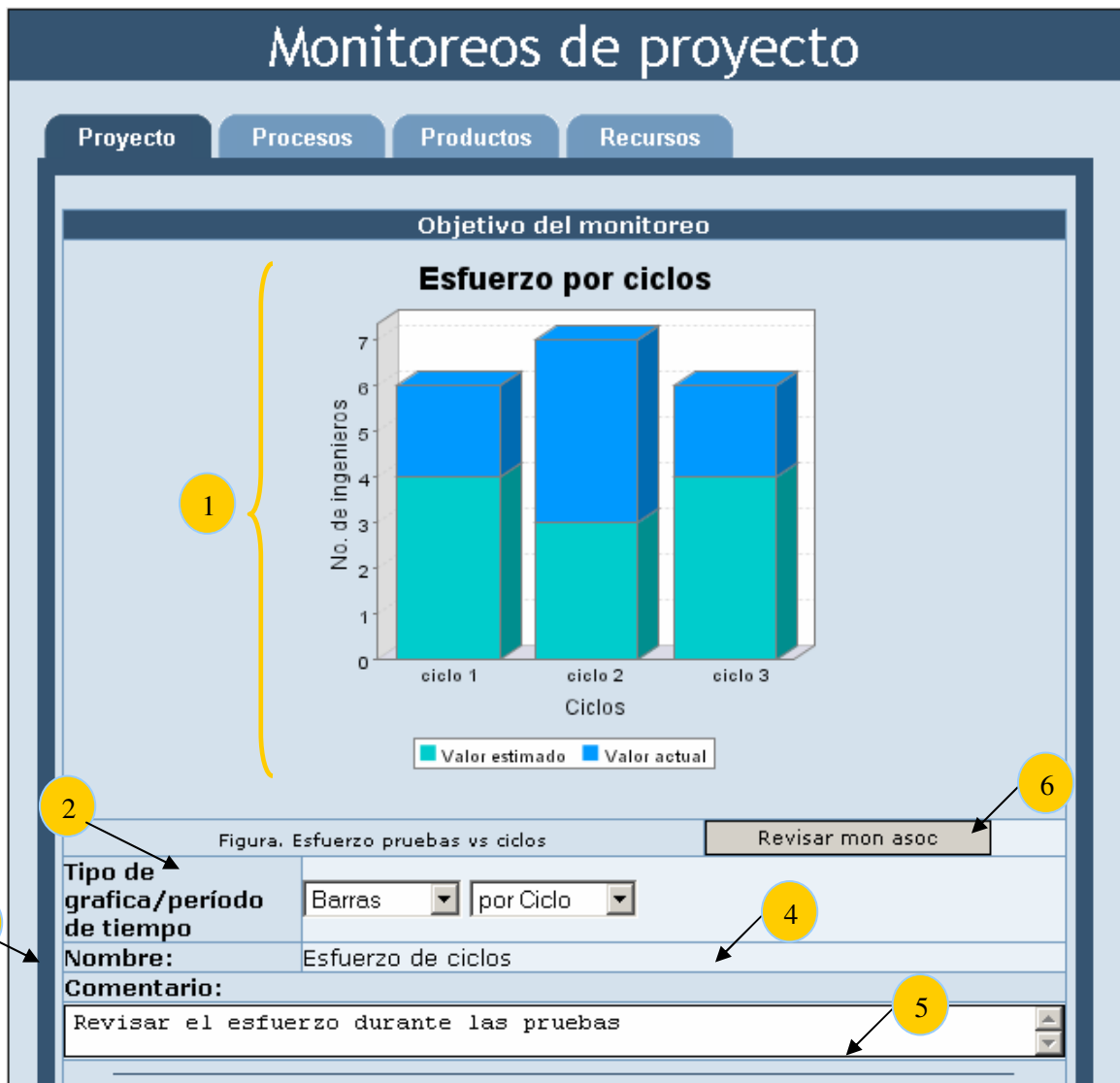


Figura 42. Datos del monitoreo.

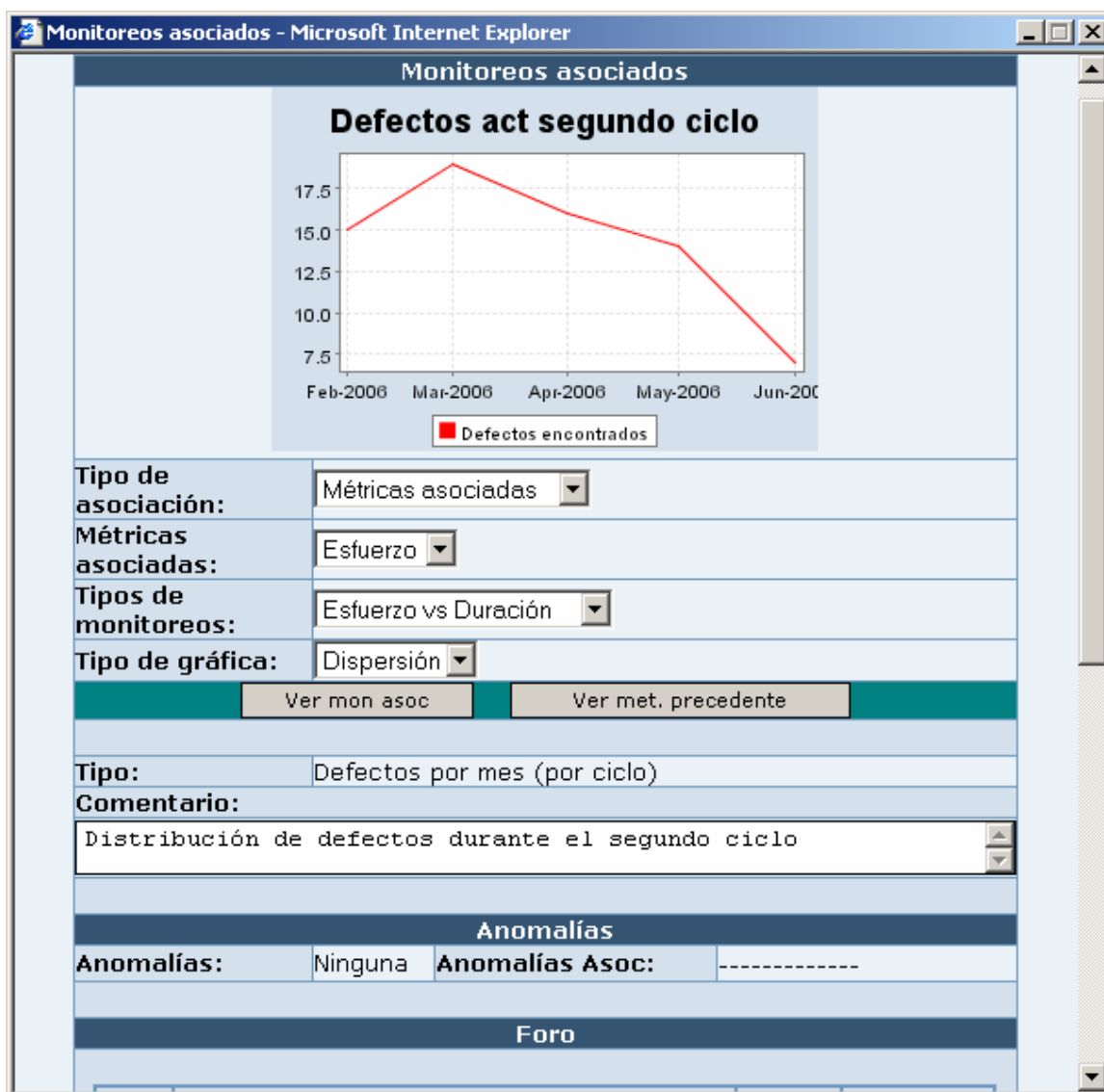


Figura 43. Monitoreos asociados.

En la segunda sección se presentan los detalles de las anomalías, estas se presentan en un listado (Figura 44 (1)) y para cada anomalía seleccionada se muestra la descripción de esta (Figura 44 (2)). También se presentan las anomalías históricas asociadas al tipo de monitoreo (Figura 44 (3)) y las anomalías generales. Al seleccionar alguna anomalía histórica se presentan los detalles de esta (Figura 44 (4)). A través del botón, *convocar discusión grupal* (Figura 44 (5)), es posible citar a la discusión para la revisión de las anomalías o de los datos del monitoreo.

Figura 44. Anomalías.

Las secciones de interpretación y retroalimentación (secciones 3 y 4, respectivamente) se describen mas adelante.

La pantalla de los detalles de los monitoreos (Figura 45) tiene el botón de *Registrar información* para actualizar la configuración del monitoreo, cuando se le da clic a este botón se hace la petición en el servidor y este actualiza los cambios en la base de datos para después presentar la pantalla del calendario (Figura 37).

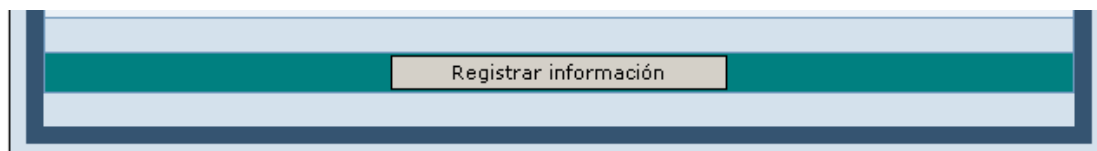


Figura 45. Botón para Registrar información.

V.3.6. Interpretar monitoreo

La pantalla para la interpretación de los monitoreos es la GUI de los detalles del monitoreo, el acceso es el mismo que para Revisar Monitoreos (V.3.5).

La sección para la interpretación esta en función de cada anomalía, en otras palabras, mientras las anomalías representan los efectos, las causas están dadas por las circunstancias, es decir, por el resultado de la interpretación.

El objetivo que tiene el ingeniero de software es encontrar indicios de porque se presenta un valor anómalo, para esto puede hacer uso de interpretaciones históricas, explorar las interpretaciones de los monitoreos asociados o verificar las recomendaciones del sistema.

Para las interpretaciones históricas se obtienen aquellas que son del tipo de la anomalía que se esta revisando o bien las interpretaciones generales, este tipo de interpretaciones se reflejan en el listado de interpretaciones históricas (Figura 46).

Interpretacion	
[Circunstancias]	
Anomalías:	Incremento de Ing ▼ Tipo: Asoc anomalía ▼
Circunstancias Historicas ing p mal ent req Ent deficiente >> <<	
Nombre:	ing p mal ent req Nueva circunstancia
Descripción:	Ing. de pruebas entendio mal req + Info Histórica
Informacion adicional:	Staff inexperimentado (-) [Kitchenham, 1988, ref literaria] Staff pobremente motivado (-) [Kitchenham, 1988, ref literaria] Requerimientos bien entendidos (+) [Kitchenham, 1988, ref literaria]
Referencia:	Empirica [proy. Configurador de Graf, 2004]
Tipo de retroalimentación:	Desfavorable

Figura 46. Interpretación.

Cada interpretación tiene asociada información adicional, cuando se selecciona una interpretación entonces se carga la información adicional en (Figura 47), si se usa una interpretación histórica, es decir, si se transfiere a través del botón ">>", entonces también se puede asociar la información adicional correspondiente.

[Información adicional]		
Monitoreo:	Favorable	
Información adicional Staff altamente motivado >> <<		
Nombre:	Staff alt mot	Nueva info adicional
Descripción:	Staff altamente motivado	+ Info Histórica
Referencia:	---	

Figura 47. Información adicional.

Las recomendaciones del sistema se presentan en un listado de recomendaciones y solo se muestran las que están asociadas al tipo de gráfica. El ingeniero transfiere de la caja de recomendaciones disponibles (usando el botón ">>") a la caja de recomendaciones que le sirven para obtener conclusiones (Figura 48).

[recomendaciones]		
Tipo de información:	Esfuerzo de pruebas por ciclos	
Recomendaciones Historicas Indic. cp retrab >> <<		
Nombre:	Indic. cp retrab	Nueva referencia
Descripción:	Si se da un esfuerzo alto, entonces revisar el número de casos de prueba para una indicación del retrabajo	+ Info Histórica
Referencia:	[Jarmighson, 1992, ref. literaria]	
Foro:	[No hay foro]	Convocar disc grupal

Figura 48. Recomendaciones.

También se tiene la revisión de los monitoreos asociados, para acceder esta pantalla se pueden abrir los monitoreos asociados y se verifican las interpretaciones de las anomalías que se presentaron en el monitoreo asociado (Figura 43).

Para convocar a una discusión grupal para la interpretación de los valores anómalos se le da un clic al botón (Figura 48 (1)) *Convocar disc grupal*. La pantalla para la convocatoria grupal se presenta en una nueva ventana (Figura 49). Esta pantalla contiene el nombre de la convocatoria (Figura 49 (1)), los participantes del equipo (Figura 49 (2)) y las fechas de inicio (Figura 49 (3)) y cierre de la discusión (Figura 49 (4)), también se presenta a que elemento del proceso de medición (Figura 49 (5)) esta asociada a la convocatoria de la discusión. Para enviar la convocatoria grupal se presiona el botón *Enviar* (Figura 49 (6)) e inmediatamente se envía la petición al servidor y este genera una entrada en el foro de discusión y envía correos electrónicos a los participantes.

The screenshot shows a web browser window titled 'Convocatoria discusión - Microsoft Internet Explorer'. The form itself has a title bar 'Convocatoria discusión'. It contains the following elements:

- 1. Nombre de la discusión:** *Esfuerzo bajo*
- 2. Fase de medición:** *Interpretación*
- 3. Seleccione a elementos con experiencia en interpretaciones similares:** This section contains a list of participants: Pepe Arevalo, Vane Lopez, Luis Lobo, and Ana Lopez. There are also buttons for adding (>>) and removing (<<) participants.
- 4. Descripción de la discusión:** *Se ha registrado un esfuerzo bajo en las pruebas*
- 5. Fecha y hora de inicio de la discusión:** A date and time selector showing 'Hora 2 pm :00'.
- 6. Fecha y hora de cierre de la discusión:** A date and time selector showing 'Hora 2 pm :00'.
- Enviar:** A button at the bottom right of the form.

Figura 49. Convocatoria discusión.

Una vez que el ingeniero de software ha revisado el contexto de la interpretación y tiene conclusiones, entonces puede agregar comentarios al foro de discusión.

V.3.7. Revisión retroalimentación

El funcionamiento de la retroalimentación es muy parecido a la interpretación (V.3.6), este se ubica dentro de la pantalla de los detalles del monitoreo (Figura 42) y el acceso a este módulo es similar al que ocurre para la interpretación de un monitoreo.

Antes de obtener una conclusión sobre alguna acción de retroalimentación es útil revisar el listado de las acciones históricas de retroalimentación (Figura 50 (1)), estas acciones están en función de las circunstancias (Figura 50 (2)) que se han presentado.

Ya que se tienen los fundamentos para dar un comentario sobre la retroalimentación, entonces se escribe esté (Figura 50 (3)) y se especifica si es una respuesta a corto o largo plazo, después se da un clic en el botón de <enviar> para registrar la acción de retroalimentación.

Figura 50. Retroalimentación.

Al igual que la interpretación donde existe una entrada en el foro de discusión para conocer cual es la interpretación mas acertada, para discutir cual es la retroalimentación mas precisa también se cuenta con un foro de discusión.

La otra tarea de la retroalimentación es establecer cuales serán las métricas a tener especial atención para el siguiente ciclo de desarrollo, para esto se le da clic al botón de *métricas a monitorear* (Figura 50 (4)) y se abre la ventana con el catálogo de métricas de SARE (Figura 51). Para la asignación entre las métricas y las acciones de retroalimentación primero se transfiere la métrica como una de las mediciones a seguir para el siguiente ciclo de desarrollo (1) y se selecciona una acción de retroalimentación del listado de acciones (2), la métrica y la acción quedan asociadas cuando se le da clic al botón de *Asignar* (3)

Métricas a monitorear - Microsoft Internet Explorer

Configuración de mediciones

Elemento de la ingeniería de software: Proyecto

Subelemento de la ingeniería de software:

☐ Habilitar las mediciones para todos los subelementos

Mediciones

Tam
Esfuerzo
Duracion
Costo
Productividad

Asignadas al proyecto

Asignar

Acciones de retroalimentacion

Req mas completos
Pers mas entrenado

Figura 51. Métricas a monitorear.

Para concluir el proceso de medición, existe una recopilación de todas las decisiones que se han tomado para el análisis de medición, esto sirve para hacer un postmorten del proceso de medición. Para acceder a está recopilación se le da un clic al botón *ruta de la*

retroalimentación (Figura 50 (5)) e inmediatamente se abre una ventana (Figura 52) donde se presentan las decisiones que se han tomado.

Ruta de retroalimentación		
Anomalía (s):	Incrementa ingenieros	✓
	↓	
Causa (s):	Entendieron mal requerimiento	✗
	Entrenamiento deficiente	✓
	↓	
Información adicional:	-----	
	↓	
Recomendación (es):	-----	
	↓	
Acciones retroalimentación:	Personal + entrenado	✓
	↓	
Conclusiones:		
Hubo mejora :	☐ Si ☐ No	
Cerrar		

Figura 52. Ruta de retroalimentación.

V.3.8. Utilerías

A continuación se presentan dos utilidades dentro de SARE: (1) Foro de discusión y (2) Mi cuenta.

El Foro de discusión se accede a través de la pestaña de primer nivel *notificaciones* y al seleccionar la pestaña de segundo nivel *Foro de discusión* dentro del contenedor central (Figura 53), el sistema presenta listados de todas la actividades que tengan asociado algún foro de discusión activo (1).

SARE Sistema de Apoyo al pRoceso de mEdición

Administración Mediciones Monitoreos Notificaciones Mi Cuenta

Equipo 1
En línea
Pepe Arevalo Lider

No conectado
Ana Lopez Analista, Diseñador
Vane Lopez Analista
Luis Lobo implementación

Mi status
Id parevalo
Nombre Pepe Arevalo
Rol Equipo 1: Lider
Proyecto actual info. del proyecto
Cambiar de Configurador de esqu
Cambiar de proyecto
Cambiar de ciclo ciclo 1

Foro de Discusión
Anuncios Foro de discusion Calendario

Pendientes
Fase proceso de medición : Recolección
Actividad: Estimar Métrica
Elemento: Proceso
Entidad Pruebas

Discusiones

	Hilos	Autor	Fecha
	Consideracion de parametros	alopez	2006-04-01 02:05:23
	Hay que definir el modelo	alopez	2006-06-01 12:25:12
	Si, el ult. proy. fue nuestra mejor est.	alopez	2006-08-01 08:21:12
	Yo me voy por COCOMO2, puntos de objet.	vlopez	2006-07-01 14:23:01
	Sugiero MERMAID o COCOMO2	parevalo	2006-07-01 09:45:02

Centro de Investigación Científica y Educación Superior de Ensenada, 2006

Local intranet

Figura 53. Foro de discusión.

La utilidad de Mi cuenta se accede al seleccionar la pestaña de primer nivel *Mi cuenta* y lo que se presenta es una GUI donde el ingeniero de software puede cambiar su información personal (Figura 54), una vez que modifica su información personal le da un clic al botón *Enviar* y se actualizarán en la base de datos.

SARE: Sistema de Apoyo al pRoceso de mEdición - Microsoft Internet Explorer

File Edit View Favorites Tools Help

SARE Sistema de Apoyo al pRoceso de mEdición [Salir](#)

Administración Mediciones Monitoreos Notificaciones **Mi Cuenta**

Equipo 1	
En línea	
Pepe Arevalo	Lider
No conectado	
Ana Lopez	Analista, Diseñador
Vane Lopez	Analista
Luis Lobo	implementación

Mi status	
Id	parevalo
Nombre	Pepe Arevalo
Rol	Equipo 1: Lider
Proyecto actual	info. del proyecto
Cambiar de proyecto	Configurador de esqu
Cambiar de ciclo	ciclo 1

Información personal

Cuenta

Actualización de Contraseña

Contraseña anterior:

Contraseña nueva:

Repetir contraseña nueva:

Centro de Investigación Científica y Educación Superior de Ensenada, 2006

Local intranet

Figura 54. Mi cuenta.

En el siguiente capítulo se presentan las pruebas sobre el sistema SARE, estas pruebas sirven para verificar que se hayan cubierto los requerimientos del sistema (capítulo IV).

Capítulo VI.

Pruebas al sistema SARE

VI.1. *Introducción*

Las pruebas del software son un elemento crítico para la garantía de la calidad del software. Son parte integral del proceso de desarrollo de software y del proceso del aseguramiento de la calidad en sistemas. Las pruebas representan una revisión final para conocer si el producto cubre con las necesidades que originaron su desarrollo. Durante el proceso de pruebas también se detectan la mayoría de los defectos y las anomalías del sistema.

En este capítulo se presentan las pruebas aplicadas al sistema SARE. Básicamente se cubren las pruebas de aceptación, estas verifican que SARE cubra los requerimientos (capítulo IV) que forman la base de una guía del proceso de medición para equipos de trabajo distribuidos.

VI.2. *Diseño de las pruebas*

Para realizar las pruebas es necesario hacer un diseño efectivo de estas, este diseño debe cubrir los aspectos más importantes de SARE. Con un buen diseño de pruebas, se puede concluir con certeza si SARE cumple o no con los requerimientos planteados y por otro lado también se obtiene una indicación de cual es la calidad del sistema.

Para realizar las pruebas de aceptación se requiere un plan de pruebas y los datos de prueba. En el plan de pruebas se especifican los casos de prueba, cada uno de estos puede cubrir uno o más requerimientos funcionales, para realizar el plan de pruebas se necesitan datos, a estos datos se les conoce como datos de prueba.

El plan de pruebas en SARE se enfoca a un escenario donde se cuenta con dos líderes, un líder del proyecto y un líder del departamento de pruebas, para elaborar estos casos de prueba se consideraron escenarios que comúnmente se presentan en la empresa en la que estos ingenieros laboran.

VI.3. *Pruebas de aceptación*

Al terminar el desarrollo del sistema o módulos del mismo, es posible realizar las pruebas de aceptación o de validación. La validación significa que el sistema cubre las necesidades descritas en el documento de requerimientos.

A cada uno de los casos de prueba de aceptación también se le conoce como pruebas de caja negra, para realizar estas pruebas se proveen las entradas para su ejecución, se plantean una serie de escenarios de la prueba y por ultimo se registran los resultados de la misma.

Al terminarse la ejecución de cada caso de prueba de aceptación, se tienen dos posibles salidas: (1) que las características de funcionamiento o de rendimiento están de acuerdo con los requerimientos y sean aceptables o (2) que se descubran una desviación de las especificaciones y se cree una lista de estas deficiencias.

VI.3.1. Objetivos y factores de motivación de las pruebas

Se ha hablado de que las pruebas de aceptación buscan confirmar que el sistema cubra exitosamente los requerimientos de software. Sin embargo pueden ser varias las razones por las que el sistema no cumpla con las expectativas, a continuación se presentan los principales defectos que pueden afectar la funcionalidad de SARE:

1. La interfaz grafica no represente la funcionalidad requerida por SARE.
2. No exista consistencia en la información que se registra en la base de datos.
3. No se activen las notificaciones en el momento en que un elemento del desarrollo o del proceso de medición tenga cambios y estos impacten el valor de alguna medición.
4. Las consultas sobre los datos históricos (memoria organizacional) no tengas los datos correspondientes.
5. No se mantenga un adecuado control de la configuración de las versiones, es decir que la información de una versión este asociada a otra versión.

VI.3.2. Metodología

Para realizar las pruebas de aceptación es necesario realizar una serie de pasos secuenciales, si en alguno de estos pasos existen deficiencias se tiene el problema de que las pruebas en vez de contribuir a incrementar la calidad del sistema, acentúan los defectos y estos se hacen aún mas difíciles de encontrar.

Los pasos para las pruebas de aceptación son los siguientes:

- Tener un conjunto de datos de prueba que sirva de entrada para los casos de prueba.
- Diseñar el plan de pruebas (donde cada caso de prueba se compone de un nombre, objetivos a cumplir y resultados de la prueba).
- Aplicar el plan de pruebas.
- Documentar el resultado de cada prueba. A continuación se describen cada uno de estos pasos.

VI.3.2.1. Desarrollo de datos de pruebas

Los datos de prueba constituyen las entradas de información para verificar que SARE cumpla con los requerimientos funcionales.

Para la obtención de los datos de prueba se contó con la descripción que dieron los líderes (que probaron SARE, ambos líderes pertenecen al mismo equipo de desarrollo) referentes a los escenarios, las necesidades que se presentan en dichos escenarios se describen a continuación.

1. Los líderes de equipos coordinan ingenieros de software para que estos cumplan con sus tiempos de desarrollo (también conocidos como tiempos de entrega). Los datos de prueba que se presentan principalmente son:
 - a. Un proyecto de software se divide en ciclos de trabajo, se tiene una duración fija para el proyecto y la duración de cada ciclo de trabajo puede variar.
 - b. Para los equipos de trabajo, se tiene un analista, un equipo de desarrollo y un equipo de pruebas, cada uno de los equipos tiene su respectivo líder.
2. La función de los líderes es monitorear cual es el progreso y la calidad con que se implementan los nuevos requerimientos.
3. Aunado a esto también quieren reaccionar a cualquier problema que se presente en el proyecto y conocer de forma precisa si es factible implementar los requerimientos según la planeación inicial que se había hecho.

Las métricas que se desean verificar son principalmente la duración y la calidad. Por otro lado, cada equipo de trabajo esta distribuido y se desea que con la guía SARE y la utilización de la herramienta que acompaña la guía, se pueda llevar registro de las repercusiones de cada cambio o decisión que se va tomando en el proyecto.

VI.3.2.2. Diseño del plan de pruebas

El plan de pruebas es un documento que describe el como se llevaran a cabo las pruebas de aceptación.

La base del plan de pruebas son los casos de pruebas, en el plan de pruebas se especifica como los casos de prueba cubren a los requerimientos.

Para la definición de los casos de prueba es necesario responder a las preguntas de ¿Qué se va a probar?, ¿Cómo se harán las pruebas?, ¿Qué resultados se espera alcanzar? y ¿Cuáles son los resultados que se obtienen?

A continuación se presenta un listado de los casos de prueba para SARE:

1. Acceder al sistema
2. Agregar proyecto
3. Agregar equipos de trabajo
4. Agregar elemento de la IS

5. Cambiar a contexto de desarrollo
6. Obtener datos métricas asociadas
7. Formular métricas
8. Hacer estimación
9. Registrar valor actual
10. Revisar Monitoreo
11. Revisar Interpretación
12. Revisar Retroalimentación
13. Programar eventos
14. Participar en discusión
15. Verificar sistema de notificaciones

Con estos casos de prueba se cubren los requerimientos de SARE, en la Tabla III se presenta la relación entre los requerimientos y los casos de prueba.

Tabla III. Relación entre requerimientos funcionales y casos de prueba.

Requerimiento funcional	Número de caso de prueba
Acceso	
Control de acceso	1
Administración	
Registro del proyecto y de los ciclos de trabajo	2
Registro de equipos, ingenieros y roles	3
Registro de productos, procesos de la IS	4
Control de cambios para los elementos de la IS	4
Cambiar de proyecto	5
Formulación	
Configurar métricas	7
Asignación de responsables.	7
Recolección	
Establecer estimaciones	6,14
Configurar estimaciones de métricas asociadas	6, 13
Actualizar reestimaciones	8, 14,15
Calcular el valor actual	9,15
Estimaciones individuales o grupales	8, 14, 15
Análisis	
Gráficas predefinidas	10
Establecer valores de referencia	10
Configurar revisión del monitoreo	13
Configurar objetivo del monitoreo	10
Configurar monitoreos de las métricas asociadas	6, 13
Revisar monitoreos asociados	6, 10
Interpretación	
Lista de circunstancias	11
Discusiones de la interpretación	11, 14
Recomendaciones del sistema	11

Tabla III (cont). Relación entre requerimientos funcionales y casos de prueba.

Requerimiento funcional	Número de caso de prueba
Retroalimentación	
Lista de acciones de Administración	12
Discusiones de la retroalimentación	12, 14
Métricas a monitorear	7
Registro de la ruta de la retroalimentación	12

A continuación se describe como se llevaron a cabo estos casos de prueba.

VI.3.2.3. Aplicación de las pruebas

A continuación se describen los objetivos para cada caso de prueba, estas pruebas se realizaron considerando un escenario real del uso de SARE, la elección del escenario no resulto compleja dado que la guía SARE es genérica a nivel de administración de equipos y se aplica a cualquier desarrollo de software, fue fácil motivar a un equipo de trabajo para que hicieran pruebas de aceptación del uso que le podrían dar a SARE.

Prueba 1. “Acceder al sistema”

Esta prueba tiene como finalidad verificar el control de acceso a SARE, todos los usuarios de SARE deben de pasar por este filtro y es importante que funcione adecuadamente para así evitar al máximo que gente no autorizada, este haciendo mal uso de la información.

Los objetivos de esta prueba son:

- Un usuario válido introduce una contraseña incorrecta y el sistema le niega el acceso.
- El usuario *administrador* no selecciona proyecto alguno de la pantalla de acceso e ingresa al sistema.
- El usuario *administrador* selecciona un proyecto de la pantalla de acceso e ingresa al sistema.
- El usuario *ingeniero de software* ingresa al sistema y obtiene la información del proyecto seleccionado en la pantalla de acceso al sistema.
- El usuario accede a los recursos según el rol que tiene asignado.

Requisitos y restricciones

Se requiere conocer en nombre de usuario y contraseña del usuario, así como el proyecto con el cual se quiere ingresar al sistema.

Descripción de la prueba

Esta prueba consiste de cuatro posibles escenarios:

- Escenario 1. Usuario no registrado en ningún proyecto y que pueda acceder a SARE.
- Escenario 2. Usuario se encuentre registrado en un proyecto y pueda ingresar al sistema.
- Escenario 3. Usuario registrado pueda acceder a un proyecto al que no tiene acceso.
- Escenario 4. Un usuario ingresa al sistema y ve información de la que no esta autorizarlo el verla.

Resultados esperados

Ante la ocurrencia de cada uno de los cuatro escenarios indicados, se presentan las siguientes alternativas:

- Identificador del usuario incorrecto y contraseña invalida, entonces se despliega mensaje de error.
- Identificador de usuario incorrecto entonces se despliega mensaje de error.
- Identificador y contraseña del usuario son permitidas, entonces el sistema muestra la información según el rol del usuario.

Prueba 2. “Agregar proyecto”

Esta prueba tiene como objetivo dar de alta o configurar un proyecto de software, dicho proyecto sirve para que ingenieros de software se agreguen a un proyecto en particular.

Los objetivos para agregar un proyecto son:

- Agregar nombre y descripción al proyecto.
- Clasificar el proyecto en alguno de los tipos o subtipos existentes.
- Agregar limitaciones de esfuerzo, costo y duración al proyecto.
- Agregar ciclos de trabajo.
- Verificar si el proyecto se agrego exitosamente.
- Verificar si el usuario con el rol de líder esta habilitado para agregar un proyecto.
- Verificar que el rango de los ciclos de trabajo no exceda el inicio y fin del proyecto.

Requisitos y restricciones

El único usuario que debe estar permitido para la creación del proyecto es aquel que tenga el rol de líder.

Descripción de la prueba

El usuario selecciona la sección de Administración y luego proyecto, para iniciar la creación de de un proyecto, se pueden ingresar los datos de proyecto definidos en el apéndice B de datos de las pruebas. También, se configuran los ciclos de trabajo y se envían los datos al servidor para verificar que todo se haya almacenado correctamente.

Resultados esperados

- Verificar que el proyecto se registre en la base de datos.

- Verificar la disponibilidad inmediata del proyecto que se desea acceder.

Prueba 3. “Equipos de trabajo”

Esta prueba sirve para verificar la flexibilidad que se persigue dar a los equipos de trabajo dentro de SARE. La flexibilidad esta en función del crear un numero ilimitado de equipos de trabajo que pueden ubicarse en cualquier nivel.

Los objetivos de este caso de prueba son:

- Agregar datos de los equipos de trabajo.
- Agregar datos de los ingenieros.
- Agregar roles para cada ingeniero de software.
- Verificar que el usuario *administrador* pueda agregar usuarios al sistema.
- Verificar que el usuario *líder* pueda agregar usuarios al sistema.
- Verificar que el usuario este asociado al proyecto, rol y equipo(s) de trabajo que le fueron asignados.

Requisitos y restricciones

El usuario con los privilegios necesarios para hacer uso de este modulo es el líder de equipos.

Descripción de la prueba

El usuario selecciona la sección de Administración y luego la opción equipos de trabajo y se presenta la pantalla de visualización si es que ya existen equipos, de no ser así se presenta la pantalla de dar de alta un equipo.

Lo primero que se hace es verificar que se puedan agregar equipos e ingenieros de software por separado y después poder seleccionar un equipo y poderle asignar ingenieros. También se pueden definir los roles de los ingenieros. En el apéndice de los datos de prueba se muestran diversos equipos de trabajo.

Resultados esperados

- Verificar que solo el líder de equipo o del proyecto pueda crear nuevos equipos.
- Verificar la consistencia de la información entre los roles que tiene un ingeniero y también revisar que este asociado al equipo que originalmente se le asigna. Verificar que los equipos de trabajo pertenezcan al proyecto para el que fueron creados.

Prueba 4. “Agregar elemento de la IS”

Esta prueba sirve para verificar que las entidades del contexto administrativo se manejen de forma consistente, ordenada y flexible.

Los objetivos de agregar elemento de la IS son:

- Que el ingeniero encargado sea quien este habilitado a actualizar el elemento.
- Que la actualización genere una nueva versión del elemento de la IS.
- Que cuando haya un cambio sobre el elemento, esto se refleje en los cambios sobre las estimaciones
- Que si un elemento se actualiza, entonces se activa una notificación para indicar que las estimaciones de los elementos asociados o los elementos mismos deben verificarse.

Requisitos y restricciones

El usuario con los privilegios necesarios puede manipular las entidades del contexto administrativo.

Descripción de la prueba

El usuario selecciona la sección de Administración y después la opción de Requerimientos, Módulos o Pruebas, para cada una de estas opciones el usuario accesa a un listado de elementos, a partir de ahí, si es que tiene los privilegios, entonces puede seleccionar y borrar algún elemento del listado, así también puede crear uno nuevo.

Cuando actualiza un elemento del desarrollo puede asociar a dicho elemento otros elementos, es decir, puede verse influenciado o relacionado a considerar otro elemento cuando es que sucedan cambios en el elemento actual.

En los datos de prueba se tienen requerimientos, así como módulos de diseño e implementación y casos de prueba, se tienen principalmente referencias entre requerimientos y módulos de implementación.

Resultados esperados

- Verificar que solo los usuarios con los roles permitidos sean los que pueden manipular las entidades contextuales.
- Verificar la consistencia entre la información que se actualiza y la de la base de datos, también verificar que la GUI refleje los datos actuales de la base de datos.
- Verificar las asociaciones entre los elementos y revisar que la actualización de un elemento asociado impacte los elementos del desarrollo respectivos.

Prueba 5. “Cambiar a contexto de desarrollo”

Esta prueba es útil para verificar que la información contextual se despliegue y le sirva al usuario para discernir cual es el ambiente de desarrollo actual con respecto a proyectos o ciclos anteriores.

Los objetivos de cambiar el contexto de desarrollo son:

- Verificar que cuando se cambie el ciclo de desarrollo del proyecto actual, o se cambie de proyecto, entonces se cambie el contexto de la página actual al ciclo o al proyecto seleccionado.

- Verificar que cuando se cambie de proyecto entonces se cargue la información del primer ciclo del proyecto seleccionado.

Requisitos y restricciones

El usuario solo puede ver la información de los proyectos en los que ha participado.

Descripción de la prueba

El usuario selecciona del menú de izquierda alguno de los listados de los proyectos y de ciclos anteriores, con esta información el usuario debería cargar el contexto del ciclo o proyecto actual seleccionado y navegar entre la información de dicho proyecto, es decir, moverse en el tiempo.

Resultados esperados

- Verificar que solo se cargue la información de proyectos o ciclos anteriores en los que haya participado el usuario
- Verificar que cuando se cambia alguno de los proyectos o ciclos que ya han ocurrido, entonces para regresar al proyecto actual, este aparece en el mismo listado de proyectos y ciclos.

Prueba 6. “Obtener datos métricas asociadas”

Esta prueba sirve para verificar que las métricas asociadas que ofrece SARE estén en función de aquellas que dio de alta el usuario.

Los objetivos al obtener las métricas asociadas son:

- Verificar que se cargue la información de los monitoreos asociados
- Verificar que las métricas de los monitoreos asociados se hayan dado de alta en la etapa de formulación.

Requisitos y restricciones

El usuario solo puede contar como métricas asociadas aquellas métricas que aparecen en un monitoreo asociado y que además las haya registro en la etapa de formulación.

Descripción de la prueba

El usuario selecciona la opción de “Mediciones” y a partir de ahí puede hacer una estimación de algún métrica, una vez que se hace dicha estimación, el sistema le ofrece la opción de también configurar métricas que comúnmente se asocian a la métrica que estimo el usuario.

Resultados esperados

- Verificar que de las métricas asociadas, solo se presenten aquellas que haya dado de alta el usuario

Prueba 7. “Formular métricas”

Esta prueba corresponde a la primera fase del proceso de medición y se debe verificar que las mediciones que se den de alta sean aquellas que efectivamente se les va a dar un seguimiento, también sirve para revisar que los ingenieros queden como responsables de una medición.

Los objetivos de formular métricas son los siguientes:

- Que la activación múltiple para los subelementos de un elemento de la IS funcione.
- Que se asignen ingenieros responsables de las mediciones.
- Que dos ingenieros no puedan quedar como los responsables de una métrica.
- Que de forma predeterminada se vayan asignando responsables.

Requisitos y restricciones

Se requiere el rol de líder del equipo ya que es el responsable de definir las mediciones a dar seguimiento.

Descripción de la prueba

El usuario selecciona la opción de “Administración” y después la opción de “Mediciones”, mediante la interfaz de formulación de métricas, el usuario puede cargar diferentes catálogos de métricas según la selección que se haga del elemento de la IS, también puede seleccionar múltiples métricas para una entidad de la IS. Una vez que se hace estas selecciones entonces se puede asignar responsables a las mediciones mediante el listado de ingenieros de software.

Resultados esperados

- Verificar que solo el líder de un equipo sea quien es capaz de dar de alta mediciones.
- Verificar que la asignación múltiple de métricas a subentidades de la IS, funcione en la GUI.
- Verificar que las mediciones se registren en la base de datos.
- Verificar que se puedan asignar los ingenieros a las mediciones y que solo un ingeniero quede de responsable.

Prueba 8. “Hacer estimación”

Esta prueba sirve para verificar que se una integración acertada entre todos los módulos que participan en la labor de hacer una Estimación.

Los objetivos hacer una Estimación son los siguientes:

- Obtener las métricas formuladas según el elemento de la IS.
- Verificar cada método de estimación y hacer uso de la memoria organizacional y de los datos históricos cuando se requiera.
- Verificar que los datos de la estimación se registran en la base de datos.

- Registrar la estimación antes de una fecha preestablecida.
- Verificar el impacto de una reestimación sobre los elementos asociados.
- Verificar que se presentan los datos del modelo si el método de estimación es por modelo.

Requisitos y restricciones

Se requiere el rol de líder o bien del responsable asignado a la medición, para poder hacer una Estimación, aunque también si se trata de una Estimación grupal y se invita a algún ingeniero, entonces este puede hacer una Estimación.

Descripción de la prueba

El usuario selecciona la opción de “Mediciones” y es cuando accesa a la pantalla principal de Recolección, solo que con el elemento de la IS, “Proyecto”. El usuario puede escoger algún otro elemento de la IS y después seleccionar las métricas que se actualizan según el elemento de la IS. Tiene opción a diferentes mecanismos de Estimación. Una vez que se selecciona una métrica y un método entonces el usuario tiene acceso a diferentes interfaces gráficas según el método.

Resultados esperados

- Verificar que se carguen las métricas según la entidad de la IS y que también se hayan dado de alta en el proyecto o en el ciclo de desarrollo.
- Verificar cada uno de los métodos de Estimación
 - Para los modelos, verificar que se despliegue la interfaz gráfica de los modelos y sus relacionados parámetros críticos.
 - Para las analogías, verificar que se desplieguen los datos históricos referentes a la Estimación que se quiere realizar.
 - Para la Estimación por grupos, verificar que se despliegue la interfaz de convocatoria grupal.
 - Para la experiencia personal, verificar que se despliegue la pantalla correspondiente.
- Verificar que el resultado de la Estimación se registre en la base de datos.

Prueba 9. “Registrar valor actual”

Esta prueba es la contraparte a hacer una Estimación y sirve para saber si hay consistencia entre el contexto que se estimó y el contexto en donde se captura el valor actual.

Los objetivos para registrar el valor actual son los siguientes:

- Que el valor actual esté asociado a la estimación a la que fue asignada.
- Si el valor actual cambia y es la base de una estimación, que se active una notificación.
- Si el cálculo del valor actual corresponde a una estimación por modelo, entonces se debe verificar que se presente la opción para calcular los valores actuales del modelo.

Requisitos y restricciones

Se requiere ser el líder o bien el rol de responsable de la métrica de la que se quiere calcular el valor actual.

Descripción de la prueba

El usuario selecciona la opción de “Mediciones” y es cuando accesa a la pantalla principal de Recolección, en dicha pantalla selecciona la opción de “Calcular valor actual”, después selecciona una Estimación de la cual quiere calcular el valor actual, registra el valor actual de esta y el sistema verifica el impacto de este valor actual.

Resultados esperados

- Verificar consistencia en que efectivamente se registre en la base de datos el valor actual de la Estimación seleccionada.
- Verificar que si se hizo una Estimación por medio de modelo y se calcula el valor actual de dicha Estimación, entonces se tenga la opción de calcular los valores actuales para los parámetros del modelo.
- Verificar el accionar del sistema de notificaciones, durante la revisión del impacto del valor actual.

Requisitos y restricciones

Se requiere ser el líder o bien el rol de responsable de la métrica de la que se quiere calcular el valor actual.

Descripción de la prueba

El usuario selecciona la opción de “Mediciones” y es cuando accesa a la pantalla principal de Recolección, en dicha pantalla selecciona la opción de “Calcular valor actual”, después selecciona una Estimación de la cual quiere calcular el valor actual, registra el valor actual de esta y el sistema verifica el impacto de este valor actual.

Prueba 10. “Revisar Monitoreo”

Verificar el mecanismo de la revisión del monitoreo y que este facilite la administración de los valores anómalos.

Los objetivos para revisar el monitoreo son los siguientes:

- Que se puedan agregar nuevas anomalías y verificar que se registren estas en la base de datos
- Verificar que se revisen las anomalías y que no pasan desapercibidas.
- Verificar que se registre en la base de datos la información que se captura para el monitoreo.
- Verificar que las métricas asociadas estén dadas de alta en la base de datos.
- Verificar la navegación entre monitoreos asociados

Requisitos y restricciones

Pueden participar todos los ingenieros del equipo, lo único es que para salvar información referente a los monitoreos, esta acción solo la puede hacer el responsable de la métrica.

Descripción de la prueba

El usuario selecciona la opción de “Monitoreos” y es cuando accesa al listado de Monitoreos, después selecciona un monitoreo del listado y ve los detalles de este, se verifican las anomalías que aparezcan en un listado y una descripción de estas, el usuario puede hacer uso de las anomalías históricas para describir la actual o bien puede generar una nueva anomalía, también puede indagar en el comportamiento de las anomalías de los monitoreos de métricas asociadas.

Resultados esperados

- Verificar que se registre en la base de datos la información que se actualiza.
- Verificar la descripción de las anomalías.
- Verificar la funcionalidad de los monitoreos asociados.

Prueba 11. “Revisar Interpretación”

Para la revisión de la interpretación se busca incorporar todos los elementos que contribuyen a hacer una interpretación que aporte causas reales acerca del comportamiento de las anomalías.

Los objetivos para revisar el monitoreo son los siguientes:

- Verificar que los datos históricos estén asociados al tipo de interpretación.
- Verificar que se mantenga la relación entre circunstancias y efectos (anomalías).
- Verificar que se presente la información adicional.
- Verificar que las recomendaciones del sistema estén asociados a la interpretación.
- Verificar que se pueda acceder a las interpretaciones de los monitoreos asociados.

Requisitos y restricciones

Pueden participar todos los ingenieros del equipo, lo único es que para salvar información referente a las interpretaciones, esta acción solo la puede hacer el responsable de la métrica.

Descripción de la prueba

El usuario selecciona la opción de “Monitoreos” y es cuando accesa al listado de Monitoreos, después selecciona un monitoreo del listado y ve los detalles de este, después accede a la sección de interpretación y revisa circunstancias históricas, también puede verificar recomendaciones de SARE, información adicional que ayuda a darle sentido a las interpretaciones, interpretaciones históricas o bien información los monitoreos asociados.

Resultados esperados

- Verificar que solo el responsable de la métrica sea el encargado de registrar información de la interpretación.

- Verificar que se registre el uso de cada uno de los elementos de ayuda para las interpretaciones.

Prueba 12. “Revisar Retroalimentación”

La revisión de la retroalimentación busca cerrar el ciclo del proceso de medición y es necesario verificar que se hayan registrados las decisiones que dan como resultado la utilidad de las métricas.

Los objetivos para revisar la retroalimentación son los siguientes:

- Verificar que los datos históricos estén asociados al tipo de retroalimentación.
- Verificar que se puedan registrar las métricas a monitorear.
- Verificar que se presente una versión resumida de las decisiones que se van tomando, a través de la ruta de la retroalimentación.
- Verificar que se mantenga la relación entre circunstancias y las acciones de retroalimentación.

Requisitos y restricciones

Pueden participar todos los ingenieros del equipo, lo único es que para salvar información referente a la retroalimentación, esta acción solo la puede hacer el responsable de la métrica.

Descripción de la prueba

El usuario selecciona la opción de “Monitoreos” y es cuando accesa al listado de Monitoreos, después selecciona un monitoreo del listado y revisa los detalles de este, después accede a la sección de retroalimentación y revisa las acciones de retroalimentación históricas, de estas acciones se registran respuestas para las causas o circunstancias y también se verifica si estas son respuestas a largo o a corto plazo, además realiza dos actividades importantes: (1) verificar la ruta de la retroalimentación y (2) configuración de las métricas asociadas a la siguiente iteración del proceso de desarrollo.

Resultados esperados

- Verificar la consistencia en la relación entre circunstancia y acción de retroalimentación, verificar que este registrada en la base de datos.
- Verificar que la ruta de la retroalimentación coincida con las decisiones que aparecen en los detalles del monitoreo.
- Verificar que se puedan registrar métricas para cuando se da el monitoreo a métricas.

Prueba 13. “Programar eventos”

Esta prueba sirve como de entrada al sistema de notificaciones, ya que los eventos que aquí se programan son los eventos que el sistema de notificaciones toma en cuenta para avisar al usuario de los eventos.

Los objetivos para la programación de los eventos son los siguientes:

- Verificar que se presenten las opciones de la programación de eventos.
- Verificar que se registre la programación de eventos seleccionada y activen notificaciones de dicha programación.

Requisitos y restricciones

Pueden participar todos los ingenieros del equipo, las actividades en donde puede ocurrir es en la programación de las reestimaciones, la del valor actual o bien la de los monitoreos.

Descripción de la prueba

Se sigue el mismo flujo que para hacer una Estimación y una vez que se hace esta se verifica que se presenten una diversidad de eventos y fechas en la cual se puede programar una actividad del proceso de medición.

Resultados esperados

- Verificar que la programación de eventos se refleje en la base de datos.
- Verificar que se presenten las opciones para la programación de los eventos.

Prueba 14. “Participar en discusión”

Esta prueba sirve para verificar que el usuario pueda comunicarse correctamente a través de la herramienta del foro de discusión.

Los objetivos para la programación de los eventos son los siguientes:

- Verificar que los comentarios estén en el límite de las fechas para el foro de discusión.
- Verificar que se den conclusiones que reflejen una decisión final sobre todo el ciclo del proceso de medición.

Requisitos y restricciones

- Dentro de un foro de discusión pueden participar todos los ingenieros que hayan recibido una invitación.
- Los comentarios en el foro deben darse antes de que se cierre este.

Descripción de la prueba

El usuario puede participar en un foro de discusión según determinada actividad del proceso de medición, por ejemplo, durante la revisión de la interpretación puede ver los últimos comentarios y si está invitado a discutir en el foro entonces tiene habilitada la opción para agregar un comentario.

Resultados esperados

- Verificar que los comentarios que introduce el usuario se almacenen en la base de datos.

- Verificar que solo se puedan dar comentarios antes de la fecha de cierre del foro.

Prueba 15. “Verificar sistema de notificaciones”

La importancia de verificar el sistema de notificaciones esta en que este sistema juega un papel fundamental en mantener ese contacto cercano que requieren las métricas del software.

Los objetivos para verificar el sistema de notificaciones son los siguientes:

- Verificar que las activaciones funcionen y envíen notificaciones inmediatamente.
- Verificar que las notificaciones se reflejen en el calendario.
- Verificar que las notificaciones se reflejen como anuncios del sistema.
- Verificar que las notificaciones por correo electrónico sean recibidas por el usuario.
- Verificar que si en el calendario una misma actividad es disparada por varios eventos, entonces en el calendario solo se muestre una actividad.
- Verificar que en los detalles de la actividad se presenten los eventos que han disparado esta.

Requisitos y restricciones

- Las notificaciones aplican para cualquier ingeniero de software que esta en la necesidad de recibir alguna notificación.

Descripción de la prueba

El usuario recibe notificaciones bajo tres vías: avisos, calendario y foro de discusión. El usuario programa una Estimación y si deseara reestimarla a la semana, entonces recibe una notificación con los días previos que haya configurado el usuario. Las notificaciones deben recibirse con el suficientemente tiempo de anticipación para aumentar así su uso.

- Verificar que las notificaciones se reciban tan pronto como se hayan enviado.

Verificar que se envíe solo la información que le corresponde a un usuario

VI.3.3. Resultados

Las pruebas sirvieron para verificar la funcionalidad de SARE y también ayudaron a detectar defectos, las pruebas se llevaron a cabo con datos cercanos a la realidad según escenarios en los que desarrollador ha presenciado en ambientes laborales. El mismo desarrollador fue el encargado de hacer las pruebas y esto asegura una mayor familiaridad con el proyecto.

El número de defectos es muy bajo, la razón de esto es que se hicieron continuas inspecciones durante las fases del proceso de desarrollo y esto evito el que se vayan pasando defectos de una fase a otra. A continuación se describen los defectos que se presentaron:

Prueba 1. “Acceder al sistema”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 2. “Agregar proyecto”

<i>Defecto</i>	No verifica que los límites de los ciclos no excedan a los límites de inicio y fin de proyecto.
<i>Solución</i>	Este error se corrigió desde el lado del cliente con JavaScript.
<i>Comentario</i>	Este defecto puede causar confusión al usuario y en un futuro los datos históricos generarían mayor confusión. Ya fue corregido.

Prueba 3. “Equipos de trabajo”

<i>Defecto</i>	Se tiene un campo donde se asigna el rol inicial del ingeniero que se da de alta.
<i>Solución</i>	Este error se corrigió al eliminar este campo, ya que pueda causar conflictos con los roles que posteriormente se pueden asignar al usuario.
<i>Comentario</i>	Ya fue corregido.

Prueba 4. “Agregar elemento de la IS”

<i>Defecto</i>	No se envía la activación para la verificación de los elementos asociados, solo para las estimaciones.
<i>Solución</i>	Se agrego la consulta correspondiente y se verifico con el modulo de notificaciones que envíe la activación a los elementos asociados.
<i>Comentario</i>	Este defecto limita al sistema de notificaciones y a pesar de no ser critico, si puede confundir al usuario respecto a la información mas reciente que pudo haber tomado de referencia. Ya fue corregido.

Prueba 5. “Cambiar a contexto de desarrollo”

<i>Defecto</i>	Al cambiar de proyecto, no se selecciona el primer ciclo, se selecciona el primero que encuentra en la base de datos.
<i>Solución</i>	Se hizo una función genérica que devuelve el primer ciclo según el identificador del proyecto.
<i>Comentario</i>	Ya se corrigió.

Prueba 6. “Obtener datos métricas asociadas”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 7. “Formular métricas”

<i>Defecto</i>	Solamente se asignan las métricas a un responsable.
<i>Solución</i>	Se agrego el listado de los ingenieros, así se puede mantener una correspondencia uno a uno, entre ingeniero y métrica.
<i>Comentario</i>	Este defecto causa exceso de trabajo a los usuarios. Ya fue corregido.
<i>Defecto</i>	Solo se habían incluido el configurar las métricas por ciclo de desarrollo
<i>Solución</i>	Se agregaron las opciones por ciclos y por proyecto, con esto se determina la vigencia de las métricas.
<i>Comentario</i>	Este defecto solo limita la flexibilidad de SARE, pero no así su funcionalidad. Ya se corrigió.

Prueba 8. “Hacer estimación”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 9. “Registrar valor actual”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 10. “Revisar Monitoreo”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 11. “Revisar Interpretación”

Defecto	No estaba incluida la información adicional en interpretación, estaba en la sección de retroalimentación.
Solución	Se cambio la información adicional a la sección de interpretación.
Comentario	Este error se detecto inmediatamente y da una indicación que la pantalla de monitoreos, interpretación y retroalimentación puede tener en un futuro un rediseño. Ya fue corregido.

Prueba 12. “Revisar Retroalimentación”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 13. “Programar eventos”

Defecto	No se había incluido la programación de los eventos para cuando se quiere programar el cálculo del valor actual.
Solución	En la página de programación de la reestimación y de los monitoreos, se agrego la opción de valor actual y las opciones de programación de eventos.
Comentario	Este defecto no limita la funcionalidad de SARE pero si le resta flexibilidad. Ya fue corregido.

Prueba 14. “Participar en discusión”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

Prueba 15. “Verificar sistema de notificaciones”

<i>Comentario</i>	No se presentaron defectos.
-------------------	-----------------------------

VI.4. Discusión de los resultados de las pruebas

Como se menciona, las pruebas de aceptación tienen como objetivo verificar que se cubran los requerimientos en SARE, el sistema cubrió ampliamente los requerimientos establecidos, sin embargo, algunos defectos que se encontraron si brindan información valiosa que sirve principalmente como retroalimentación para decisiones de diseño que se pudieron haber tomado.

VI.4.1. Pruebas de aceptación

Con los resultados de las pruebas y con los defectos que fueron corregidos, se obtuvo una herramienta con mayor estabilidad que cubre de manera fehaciente los requerimientos establecidos.

Sin embargo, tal y como lo manifestaron los líderes que adaptaron SARE a sus necesidades, hay dos factores que vale la pena discutir, el primero es referente a los indicadores gráficos que ocurren entre los elementos de la IS que mantienen referencias entre sí, posiblemente si se asignara una nueva sección donde se indique cuáles son las entidades o versiones que recientemente acaban de tener cambios y que hay otras entidad asociadas, entonces se podría tener mas seguridad en que el usuario perciba estas relaciones entre entidades de la IS. La segunda es referente a la pantalla de los monitoreos, interpretación y retroalimentación, el tener toda la información agrupada en una misma pantalla brinda el contexto asociado entre las fases del proceso de medición, sin embargo, también puede causar confusión y puede no dársele el uso correcto.

En general hace falta que usuarios reales conozcan la guía SARE y a partir de ahí hagan uso del sistema, para conocer si realmente se usarían todas las características de esta guía.

Cabe mencionar que el plan de pruebas que se implemento cubre ampliamente los requerimientos establecidos, pero también es cierto que haber hecho un plan de pruebas de integración hubiera aportado beneficios principalmente en donde los elementos de la IS se relacionan entre sí, ya sea por relaciones explícitas entre estos, o bien acciones del proceso de medición.

A continuación se presenta el último capítulo, en el cual se las conclusiones, aportaciones y trabajo a futuro de este trabajo de investigación.

Capítulo VII.

Conclusiones, aportaciones y trabajo futuro

VII.1. Conclusiones

En este trabajo de investigación se obtuvo la guía SARE, la cual sirve para la implementación de un proceso de medición enfocado a equipos de trabajo distribuidos, para ejemplificar los conceptos de esta guía se desarrollo el sistema SARE.

La motivación para este trabajo de investigación es que aún cuando las métricas de software empiezan a tener popularidad entre la comunidad de los líderes de proyectos, no así lo tiene el proceso por el que va pasando una medición y menos aun el trabajo en equipo que se puede dar dentro de este proceso.

Para el desarrollo de esta guía, se unificaron las mejores prácticas del desarrollo de software en equipos, se entró en detalle de cada una de las fases del proceso de medición y las métricas para la administración y finalmente se unificaron estas prácticas para obtener las características de SARE.

Una vez que se identificaron estas características, se procedió a un análisis formal de los requerimientos que cubren dichas características, después se continuó con las fases de diseño, implementación y finalmente las pruebas las cuales sirvieron para verificar que los requerimientos se hayan cubierto satisfactoriamente.

El resultado fue la herramienta SARE, la cual implementa todas las fases del proceso de medición y le da un nivel de detalle en cada fase que permite tener valor agregado de las métricas que aunado al contexto administrativo y la flexibilidad para crear equipos, módulos y roles, entonces se verifica que la implementación de la guía es exitosa y con esto que se han cubierto los objetivos del trabajo.

VII.2. Aportaciones

Las contribuciones de este trabajo de investigación son:

- Se definió una guía para la implementación de un proceso de medición orientado a equipos de trabajo.
- Se analizo, diseño y desarrollo una herramienta para dar soporte al proceso SARE.

Se implemento una herramienta que apoya el uso de la guía SARE.

VII.3. Trabajo a futuro

Durante el desarrollo de este trabajo de investigación se van presentando oportunidades de extenderlo y enriquecerlo, también se presentan actividades que podrían darle mayor solidez a la investigación. A continuación se describe el trabajo a futuro para SARE.

- Probar la guía SARE en proyectos reales y hacer pruebas beta para obtener cuales son las secciones que pueden causar mayor confusión a los usuarios.
- Implementar métricas sobre SARE para conocer el uso real que pueda tener.
- Agregar la habilidad de establecer hipótesis sobre el comportamiento de las métricas y depuse intentar validar dichas suposiciones.
- Poder configurar métricas personalizadas que satisfagan más las necesidades de información.
- Implementar el sistema de reportes que incluya todas las graficas e información valiosa del contexto que sea significativa.
- Colaborar con otros sistemas para que el contexto administrativo obtenga la información de sistemas existentes.
- Agregar la oportunidad de que se puedan generar mas roles dentro de SARE.

Bibliografía

Boehm, B.W. 1981. "Software Engineering Economics". Prentice Hall. Englewood Cliffs, NJ, Estados Unidos. Primera Edición, 768 p.

Conallen, J. 1999. "Building web applications with uml". Addison-Wesley Longman Publishing Co, Boston, MA, Estados Unidos. Primera Edición, 352 p.

Chillarege, R., Bhandari, I.S., Chaar, J.K. 1992 "Orthogonal defect classification: A concept for in-process measurements". IEEE Transactions on Software Engineering. 18(11): 943-56 p.

Davis, A., Overmeyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh , A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A., Theofanos, M. 1993. "Identifying and measuring quality in a Software requirements specification". Proceedings of the First International Software Metrics Symposium, Baltimore, MD. IEEE Computer Society Press. 141-52 p.

Dumke, R.R., Winkler, A.S. 1997. "CAME Tools for an Efficient Software Maintenance". 1st Euromicro Working Conference on SoftwareMaintenance and Reengineering, Berlin, Alemania. 74-81 p.

Ejiogu, L.O. 1991. "Software Engineering with Formal Metrics". QED Technical Publishing Group, Wellesley, MA, Estados Unidos. Primera Edición. 334 p.

Fenton, N.E., Pfleeger, S.L. 1997. "Software Metrics: A Rigorous and Practical Approach". PWS Publishing Company, Londres, Inglaterra. Segunda Edición. 638 p.

Fenton, N.E, Neil, M. 2000, "Software Metrics: roadmap". International Conference on Software Engineering. Proceedings of the Conference on The Future of Software Engineering, Limerick, Irlanda. 357 – 370 p.

Humphrey, W.S. 1999. "Introduction to the Team Software Process". Addison-Wesley, Boston, MA, Estados Unidos. Primera Edición. 463 p.

Humphrey, W.S. 1995. "A Discipline for Software Engineering". Addison-Wesley, Boston, MA, Estados Unidos. Primera Edición. 816 p.

Kitchenham, B.A., Pickard, L. 1987. "Towards a constructive quality model: part 1: software quality modeling, measurement and prediction". Software Engineering Journal. 2 (4): 52-66 p.

Loomis, M.E.S. 1997. "Managing Collaborations". Hewlett-Packard Labs 1997, Technical Report 971010, Palo Alto, California, Estados Unidos. 1-8 p. revisar. <http://www.hpl.hp.com/techreports/97/HPL-97-117.pdf> (marzo, 2004).

Morisio, M. 1999. "Measurement processes are software, too". Journal of Systems and Software. 49 (1): 17-31 p.

Pressman, R. 2002. "Ingeniería de software, un enfoque práctico". McCraw-Hill, Madrid, España, Quinta Edición. 599 p.

Putnam, L., Myers, W. 1992. "Measures for Excellence: Reliable Software on Time, Within Budget". Prentice Hall, New York, Estados Unidos. Primera Edición. 304 p.

Sommerville, I. 2002. "Ingeniería de Software". Addison-Wesley, México. Sexta Edición. 712 p.

Wilson, W., Rosenberg, L., Hyatt, L. 1997. "Automated analysis of requirements specification". Proc. Fourteenth Annual Pacific Northwest Software Quality Conference, Portland, Oregon. 372-379 p.

DataDrill

<http://www.distributive.com/>

Bugzilla

<http://www.bugzilla.org/>

JSP

<http://java.sun.com/products/jsp/>

MySQL

<http://www.mysql.com/>

Apéndice A. Análisis y diseño de SARE

En este apéndice se presenta un nivel mayor de detalle para la construcción de SARE. Para el análisis se presentan casos de uso y para el diseño se presenta la descripción de cada una de las clases dentro de SARE.

A.1 Casos de uso

Los diagramas de caso de uso ofrecen un panorama rápido de toda la funcionalidad de SARE. En la Figura 55 se muestra el diagrama de casos de uso general de SARE.

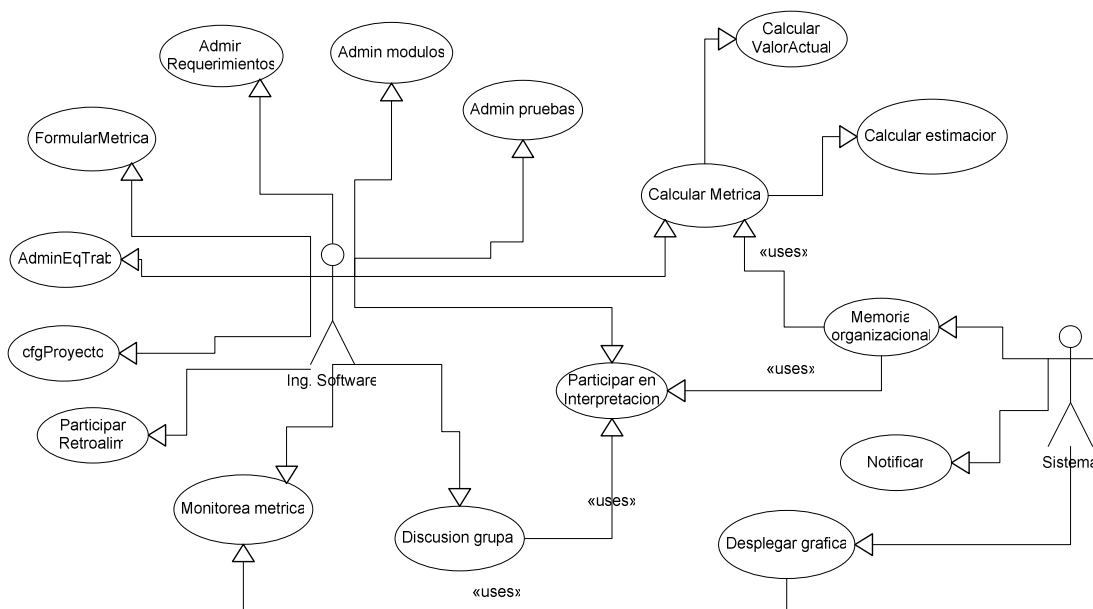


Figura 55. Diagrama general de casos de uso.

A.1.1 Formular métrica

Nombre: Formular métrica.

Descripción: Proceso para determinar que métricas se le quiere dar seguimiento.

Metas del usuario: Tener un conjunto de métricas dadas de alta.

Actores: Usuario.

Dependencias: • Es necesario que el proyecto este dado de alta en el sistema.

Acciones básicas:

7. Una vez que se lleva a cabo exitosamente el caso de uso *Ingresar al sistema*, el usuario selecciona la opción “Mediciones” y el sistema presenta la interfaz gráfica para la formulación de las mediciones, esta GUI contiene el catalogo de métricas y los ingenieros de software que serán responsables de dichas métricas.
8. El usuario selecciona las mediciones en las que esta interesado y también le asigna algún responsable, de no asignarle algún ingeniero, los responsables será

alguno de los ingenieros que tienen el rol según la métrica definida.

9. El usuario selecciona si las métricas se asocian al ciclo de desarrollo actual o bien son vigentes durante toda la duración del proyecto.
10. El usuario da de alta las mediciones en el sistema a través de la opción “enviar datos”.
11. El sistema registra los datos y presenta la pantalla donde se visualización las mediciones.

A.1.2 Participar en retroalimentación

Nombre:	<u>Participar en retroalimentación.</u>
Descripción:	Proceso para llevar a cabo la revisión de las acciones de retroalimentación.
Metas del usuario:	Conocer las acciones de retroalimentación.
Actores:	<i>Usuario.</i>
Dependencias:	• Es necesario que el proyecto este dado de alta en el sistema. • Las acciones de calcular valor actual han sido definidas. • Las acciones de monitoreo están definidas. • Las acciones de interpretación están definidas.

Acciones básicas:

1. El usuario accede al sistema.
2. El usuario introduce su nombre de usuario y contraseña en la interfase de acceso al sistema.
3. Al entrar el usuario al sistema, el usuario puede seleccionar la opción “Monitorear métrica” para ver un listado de las métricas que están corriendo, de forma predeterminada, el sistema presenta los monitoreos de las métricas asignadas al proyecto.
4. El usuario selecciona un monitoreo y el sistema presenta una pantalla con los detalles del monitoreo.
5. El usuario se sitúa en la sección de retroalimentación y puede participar en esta actividad a través de los siguientes mecanismos.
 - 5.1. El usuario puede acceder a retroalimentaciones históricas.
 - 5.2. El usuario puede revisar las retroalimentaciones de los monitoreos asociados
 - 5.3. El usuario puede convocar a una discusión grupal para las acciones de retroalimentación.
6. El usuario para la decisión final referente a las acciones de retroalimentación puede considerar alguna acción de retroalimentación histórica o bien, se agregar una nueva acción de retroalimentación.
7. El usuario puede verificar la efectividad de las acciones de retroalimentación al

asociar esta a las métricas a monitorear durante el siguiente ciclo de desarrollo.

8. El usuario a través de la opción “ruta retroalimentación” tiene acceso a un resumen referente a las decisiones que se han tomado durante el proceso de medición.

A.1.3 Configurar proyecto

- Nombre:** Configurar proyecto.
- Descripción:** Proceso para llevar a cabo la configuración de un proyecto, dentro de un contexto administrativo.
- Metas del usuario:** Tener dado de alta un proyecto.
- Actores:** *Usuario líder.*
- Dependencias:**
- Haber ingresado exitosamente al sistema.
- Acciones básicas:**
1. El rol de quien Access al proyecto debe ser aquel que disponga de un rol de líder. Una vez que se lleva a cabo exitosamente el caso de uso Ingresar al sistema, el usuario selecciona la opción “Administración” y el sistema presenta la interfaz gráfica para la captura del proyecto.
 2. El usuario introduce los datos de nombre, descripción del proyecto, tipo, subtipo del proyecto, restricciones de duración, personal y costo. También se pueden configurar los ciclos de trabajo del proyecto actual, estos no deben exceder la duración del proyecto, de lo contrario el sistema debe enviar una notificación acerca de este hecho.
 3. El usuario envía los datos al servidor a través de la opción “enviar Datos”.
 4. El sistema registra los datos y presenta la pantalla donde se visualizan los datos del proyecto.

A.1.4 Administrar equipos de trabajo

- Nombre:** Administrar equipos de trabajo.
- Descripción:** Proceso para llevar a cabo la administración de los equipos de trabajo.
- Metas del usuario:** Tener dado de alta equipos de trabajo.
- Actores:** *Usuario líder.*
- Dependencias:**
- Haber ingresado exitosamente al sistema.
 - Haber registrado un proyecto en el sistema.
- Acciones básicas:**
1. El rol de quien Access al proyecto debe ser aquel que disponga de un rol de líder. Una vez que se lleva a cabo exitosamente el caso de uso Ingresar al sistema, el usuario selecciona la opción “Equipos de trabajo” y el sistema presenta la interfaz gráfica para la captura de los equipos.
 2. En esta interfaz para la creación de los equipos de trabajo el usuario lleva a cabo

las siguientes acciones:

- 2.1. Crear equipos. Los datos que se requieren son: nombre de los equipos y la descripción de este.
- 2.2. Crear ingenieros. Los datos que se requieren son: nombre, nombre de usuario y correo.
- 2.3. Puede asignar ingenieros a equipos del proyecto simplemente cambiando la selección del listado de equipos y asignando del listado de ingenieros disponibles al listado de ingenieros asignados al proyecto.
- 2.4. Puede cambiar los roles de un ingeniero, primero selecciona un ingeniero del listado y después le habilita sus roles correspondientes.
3. El usuario envía los datos al servidor a través de la opción “enviar Datos”.
4. El sistema registra los datos y presenta la pantalla donde se visualizan los equipos de trabajo.

A.1.5 Administrar requerimientos

Nombre:	<u>Administrar requerimientos.</u>
Descripción:	Proceso para llevar a cabo la administración de los requerimientos.
Metas del usuario:	Tener dado de alta requerimientos de software.
Actores:	<i>Usuario líder de requerimientos</i>
Dependencias:	• Haber ingresado exitosamente al sistema. • Haber registrado un proyecto en el sistema.

Acciones básicas:

1. El rol de quien Access al proyecto debe ser aquel que disponga de un rol de líder de análisis o de requerimientos o líder de proyecto. Una vez que se lleva a cabo exitosamente el caso de uso *Ingresar al sistema*, el usuario selecciona la opción “Requerimientos” y el sistema presenta la interfaz gráfica del listado de requerimientos, se presentan los requerimientos y también las opciones para agregar y eliminar requerimientos.
2. Cuando el usuario selecciona un requerimiento en particular, se presenta las siguientes información del requerimiento:
 - 2.1. Listado de versiones del requerimiento. El usuario puede navegar entre los requerimientos a través de dicho listado.
 - 2.2. Datos del requerimiento: identificador, nombre, prioridad, nombre, status, equipo de trabajo asignado, iteración, etapa del requerimiento, descripción, racional del requerimiento y racional de la versión.
 - 2.3. Relaciones entre requerimientos. El usuario puede transferir requerimientos que estén asociados al requerimiento actual, esto sirve para conocer el impacto de las métricas.

3. El usuario envía los datos al servidor a través de la opción “enviar Datos”.
4. El sistema registra los datos y presenta la pantalla donde se visualizan el listado de los requerimientos.

A.1.6 Administrar módulos

Nombre:	<u>Administrar módulos.</u>
Descripción:	Proceso para llevar a cabo la administración de los módulos.
Metas del usuario:	Tener dado de alta módulos de diseño y de implementación.
Actores:	<i>Usuario líder de diseño e implantación, líder de proyecto.</i>
Dependencias:	• Haber ingresado exitosamente al sistema. • Haber registrado un proyecto en el sistema. • Haber registrado requerimientos del sistema.

Acciones básicas:

1. El rol de quien Access al proyecto debe ser aquel que disponga de un rol de líder del proyecto, diseño o de implementación. Una vez que se lleva a cabo exitosamente el caso de uso *Ingresar al sistema*, el usuario selecciona la opción “Módulos” y el sistema presenta la interfaz gráfica del listado de módulos, en este listado se presenta el tipo de módulo el cual puede ser de diseño o de implementación, también se tienen las opciones para agregar o eliminar módulos.
2. Una vez que el usuario Access un modulo, se presentan la siguiente información:
 - 2.1. Listado de versiones del modulo: El usuario puede ver los detalles de los módulos históricos.
 - 2.2. Datos del módulo: Nombre, prioridad, status, equipo de trabajo, iteración, fase, bandera de Pruebas, descripción y racional versión
 - 2.3. Relaciones con elemento de la IS. El usuario puede mantener diferentes relaciones respecto al módulo actual, estas relaciones pueden ser: relaciones con requerimientos, relaciones con submodulos y relaciones entre módulos.
3. El usuario envía los datos al servidor a través de la opción “enviar Datos”.
4. El sistema registra los datos y presenta la pantalla donde se visualiza el listado de módulos.

A.1.7 Administrar pruebas

Nombre:	<u>Administrar pruebas.</u>
Descripción:	Proceso para llevar a cabo la administración de las pruebas.
Metas del usuario:	Tener dado de alta casos de prueba.
Actores:	<i>Usuario líder de diseño e implantación, líder de proyecto.</i>
Dependencias:	• Haber ingresado exitosamente al sistema. • Haber registrado un proyecto en el sistema. • Haber registrado requerimientos del sistema.

- Haber registrado módulos de diseño e implementación.

Acciones básicas:

1. El rol de quien accesa al proyecto debe ser aquel que disponga de un rol de líder del proyecto o de las pruebas. Una vez que se lleva a cabo exitosamente el caso de uso *Ingresar al sistema*, el usuario selecciona la opción “Módulos” y el sistema presenta la interfaz gráfica del listado de pruebas, también se tienen las opciones para agregar o eliminar casos de prueba.
2. Una vez que el usuario accesa un caso de prueba, se presentan la siguiente información:
 - 2.1. Listado de versiones del caso de prueba: El usuario puede ver los detalles de versiones previas de los casos de prueba.
 - 2.2. Datos del módulo: Nombre, descripción, tipo de prueba, status, equipo, iteración y racional versión
 - 2.3. Las posibles asociaciones entre los casos de prueba son de dos tipos: (1) dependencia a otros casos de prueba y elemento de la IS asociado.
3. El usuario envía los datos al servidor a través de la opción “enviar Datos”.
4. El sistema registra los datos y presenta la pantalla donde se visualiza el listado de los casos de prueba.

A.1.8 Discutir en grupo

Nombre: Discutir en grupo.

Descripción: Proceso para llevar a cabo las discusiones entre el equipo de trabajo.

Metas del usuario: Crear un foro para propiciar las discusiones entre los integrantes del equipo.

Actores: *Usuario líder de diseño e implantación, líder de proyecto.*

Dependencias:

- Haber ingresado exitosamente al sistema.
- Haber registrado integrantes en los equipos de trabajo.

Acciones básicas:

1. Una vez que se lleva a cabo exitosamente el caso de uso *Ingresar al sistema*, el usuario cuenta con los siguientes caminos para alcanzar las discusiones en grupo.
 - 1.1. El usuario puede directamente ingresar mediante la opción *Notificaciones* y después la opción *Foro de discusión*. El usuario configura los parámetros de la fase del proceso de medición, la subactividad asociada al PM, así como el elemento y subelemento de la IS y después el usuario selecciona la opción de Convocar a discusión grupal.
 - 1.2. El usuario puede acceder a través de la pantalla de los monitoreos (ver caso de uso *interpretación*, y selecciona la opción de Convocar a discusión grupal.
2. El sistema presenta la pantalla de “Convocatoria a discusión grupal” y la

información que se presenta es: nombre de la discusión, fase de medición, actividad, listado de participantes, también se presenta un listado de los ingenieros del equipo que se desea invitar a la discusión grupal, descripción de la discusión, fecha e inicio de la discusión.

3. Una vez que el usuario completa estos datos, se envían al servidor a través de la opción “enviar Datos”.
4. El sistema presenta una pantalla de visualización de los datos del foro de discusión.

A.1.9 Notificar

Nombre:	<u>Notificar.</u>
Descripción:	Proceso para llevar la notificación de actividades al usuario.
Metas del usuario:	Crear mecanismos para informar al ingeniero la proximidad de los eventos.
Actores:	<i>Sistema.</i>
Dependencias:	• Que haya información de donde surjan las notificaciones del sistema.

Acciones básicas:

1. Una vez que el usuario ingresa al sistema tiene diferentes medios para recibir notificaciones del sistema, estos son:
 - 1.1. Si el usuario selecciona la opción “Notificaciones”, entonces se presenta la pantalla de recados donde se indican las actividades en las que se esta invitado a participar, estas se pueden filtrar por: fase del proceso de medición, actividad, elemento y entidad de la IS.
 - 1.2. Si el Usuario selecciona la opción “Notificaciones” y después la opción “Calendario”, entonces se presenta una vista del calendario por mes donde se indica en cada día las actividades programadas para el usuario que ingreso al sistema.
 - 1.3. Si el valor de una asociación (entidades del desarrollo asociadas o métricas asociadas) sufre un cambio, entonces el sistema envía un correo electrónico al usuario para informarle de esta situación.

A.2 Diagramas de clases

El enfoque que se le da en este trabajo a los diagramas de clases es el de la extensión de UML para aplicaciones web. A continuación se presentan las clases agrupadas, según la funcionalidad que representan, para cada clase se da una breve descripción de esta, de sus atributos y métodos así como de su visibilidad de estos (- para publico y + para privado).

A.2.1 Conexión a la base de datos

La conexión a la base de datos queda encapsulada por el objeto *ConexionBDBean*, sus atributos y métodos se describen en la Tabla IV.

Tabla IV. Atributos y métodos de la clase *ConexionBDBean*.

Clase: <i>ConexionBDBean</i>	
Descripción: Componente que realiza las conexiones al servidor de base de datos.	
Atributos	Descripción
<i>-driver:String</i>	Propiedad para establecer la conexión con el servidor de base de datos.
<i>-dbURL:String</i>	Propiedad que indica la ruta y nombre de la base de datos.
Métodos	Descripción
<i>+conectarBD():int</i>	Método para conectarse a la BD, regresa un entero, un 1 para indicar si la conexión es exitosa y 0 en caso contrario.
<i>+consultaSelect(consulta:String):ResultSet</i>	Método para realizar las consultas a la BD, regresa un objeto de tipo ResultSet.
<i>+consultaUpdate():int</i>	Método para actualizar o borrar datos a la base de datos, regresa un entero, donde 1 indica que fue exitosa la transacción y 0 que fallo la sentencia.

A.2.2 Notificaciones

Para las notificaciones intervienen una serie de páginas web (representadas como clases) que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen de la Tabla V a la Tabla IX.

Tabla V. Atributos y métodos de la clase en el servidor *ObtenerCalendario*.

Clase: <i>ObtenerCalendario</i> <<servidor>>	
Descripción: Generación de la página del lado del servidor que se encarga de obtener los eventos programados para las etapas del proceso de medición y de construir la página.	
Atributos	Descripción
<i>-idUserio:int</i>	Identificador del usuario que ingresa al sistema.
Métodos	Descripción
<i>+obtenerEventos()</i>	Método para formular la consulta que trae todos los eventos del mes correspondiente al actual.

Tabla VI. Atributos y métodos de la clase en el cliente *CalendarioGrafico*.

Clase: <i>CalendarioGrafico</i> <<pagina>>	
Descripción: Página que muestra el calendario con las fechas de los eventos programados que fueron generados por <i>ObtenerCalendario</i> <<servidor>>.	
Atributos	Descripción
-eventos:tablaHTML	Se presenta la información del mes.
Métodos	Descripción
+mostrarEventos()	Presenta los eventos que se tienen.
+abrirDetallesMon(idMetrica:int)	Abre una ventana que muestra la información desglosada para un día en particular.
+cambiarMes()	Método para navegar entre los meses del calendario.

Tabla VII. Atributos y métodos de la clase en el servidor *DatosActividad*.

Clase: <i>DatosActividad</i> <<servidor>>	
Descripción: Generación de la página del lado del servidor que se encarga de obtener los datos de las actividades de las fases del proceso de medición.	
Métodos	Descripción
+obtenerDatosActividad()	Método para obtener los detalles de las actividades.

Tabla VIII. Atributos y métodos de la clase en el cliente *DatosActividad*.

Clase: <i>DatosActividad</i> <<pagina>>	
Descripción: Página que muestra la información de los datos de las actividades que fue generada por <i>DatosActividad</i> <<servidor>>.	
Atributos	Descripción
-faseProcMed:combo	Presenta las diferentes fases del proceso de medición.
-actividad:combo	Presenta las actividades dentro de cada fase del proceso de medición
Métodos	Descripción
+cargarDatosAct()	Método para a partir de la información de los combos cargar nuevos datos.

Tabla IX. Atributos y métodos de la clase en el servidor *Notificador*.

Clase: <i>Notificador</i> <<servidor>>	
Descripción: Página del lado del servidor que realiza el envío de correos a los ingenieros para notificarle acerca de los eventos programados.	
Atributos	Descripción
-ingResponsables:Vector	Es un listado de los ingenieros responsables.
Métodos	Descripción
+enviarCorreos()	Método que envía correos a los ingenieros responsables.

A.2.3 Formulación de métricas

Para la fase de formulación de las métricas intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen en la Tabla X y la Tabla XI.

Tabla X. Atributos y métodos de la clase en el servidor FormulacionMetricas.

Clase: <i>FormulacionMetricas</i> <<pagina>>	
Descripción: Página para formular una métrica según el elemento de la ingeniería de software y asignarla a un ingeniero de software.	
Atributos	Descripción
- <i>métricas:combo</i>	Es un listado donde se muestran las métricas.
- <i>listadoMetricas:cajaElementos</i>	Caja elementos para asignar las métricas al ciclo o al proyecto actual.
- <i>aplicacionSubEntidad:multiseleccion</i>	Botón de multi selección que se usa para que se apliquen las métricas según las entidad de la IS
- <i>listadoIngenieros:CajaElementos</i>	Contiene los ingenieros del equipo y de los cuales se pueden obtener los responsables de las métricas.

Tabla XI. Atributos y métodos de la clase en el servidor ProcesarFormularioMetricas.

Clase: <i>ProcesarFormulacionMetricas</i> <<servidor>>	
Descripción: Página del lado del servidor que realiza la actualización de los datos de las métricas que fueron formuladas en <i>FormulacionMetricas</i> <<pagina>>.	
Atributos	Descripción
- <i>datosFormulacion: String</i>	Son todos los datos necesarios para actualizar las métricas.
Métodos	Descripción
+ <i>decodificarDatos()</i>	Método para decodificar toda la información referente a la formulación de las métricas.

A.2.4 Recolección

Para el módulo de Recolección intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen de la Tabla XII a la Tabla XXX.

Tabla XII. Atributos y métodos de la clase en el cliente Analogia.

Clase: <i>Analogia</i> <<pagina>>	
Descripción: Página que presenta en una tabla HTML los valores históricos anteriores (valores actuales) según la métrica que se quiere estimar.	
Atributos	Descripción
-listado:tablaHTML	Propiedad que se refiere a una tabla HTML donde se muestra el listado.
-analogía:multiseleccion	Propiedad que indica las cajas de selección para cada elemento de la lista.

Tabla XIII. Atributos y métodos de la clase en el servidor Analogia.

Clase: <i>Analogia</i> <<servidor>>	
Descripción: Página del lado del servidor que administra los datos de los valores históricos de una métrica.	
Atributos	Descripción
-idMétrica:int	Propiedad que se refiere a la métrica según el elemento de la IS
Métodos	Descripción
+obtAnalogiasHist()	Método donde según el id de la métrica, trae la información de proyectos y de ciclos anteriores.

Tabla XIV. Atributos y métodos de la clase en el cliente ParametrosModelo.

Clase <i>ParametrosModelo</i> <<pagina>>	
Descripción: Página que permite navegar entre los modelos y sus parámetros correspondientes.	
Atributos	Descripción
-modelo:combo	Listado que contiene los modelos de Estimación.
-parámetros:cajaTexto[]	Propiedad que indica las cajas de texto asociadas a los parámetros.
-valorActual:cajaTexto	Es la caja de texto donde se registra el valor actual

Tabla XV. Atributos y métodos de la clase en el servidor ParametrosModelo.

Clase: <i>ParametrosModelo</i> <<servidor>>	
Descripción: página del lado del servidor que administra los datos de los modelos así como los valores de sus parámetros.	
Atributos	Descripción
-idMétrica:int	Se refiere a la métrica según el elemento de la IS
Métodos	Descripción
+obtenerModelos()	Obtiene los modelos de Estimación asociados a la métrica de software.
+mostrarDatosEstimados()	Consulta los datos estimados para los parámetros.

Tabla XVI. Atributos y métodos de la clase en el cliente DatosEstimacion.

Clase: DatosEstimacion <<pagina>>	
Descripción: Página para capturar los datos de la estimación.	
Atributos	Descripción
-valormin:cajaTexto	Es la caja de texto donde se captura el valor mínimo de la Estimación.
-valormax:cajaTexto	Es la caja de texto donde se captura el valor máximo de la Estimación.
-estrategia:combo	Se refiere al tipo de estrategia puede ser de arriba a abajo o viceversa.
-racional:cajaTexto	Son comentarios del porque hacer la Estimación.
-suposiciones:cajaTexto	Son elementos que hacen una Estimación en un ambiente restringido.

Tabla XVII. Atributos y métodos de la clase en el cliente PrincipalRecoleccion.

Clase: PrincipalRecoleccion <<pagina>>	
Descripción: Página que presenta las opciones principales de la fase de recolección.	
Atributos	Descripción
-metricas: combo	Es un combo que contiene los elementos según los elementos de la IS
-metodos: combo	Es un combo donde se presentan los métodos por el cual se llevara la Estimación
-opcionesRec: botonesRadio[]	Es el conjunto de opciones que corresponde al menú de la fase de Recolección
-catalogoSubElementosIS: combo	Es un combo que corresponde a los subelementos de la IS cuando la entidad de la IS es producto o proceso.

Tabla XVIII. Atributos y métodos de la clase en el servidor PrincipalRecoleccion.

Clase: PrincipalRecoleccion <<servidor>>	
Descripción: Página del lado del servidor que procesa la información de la estimación y se redirecciona a una nueva página del lado del cliente de acuerdo a la información.	
Atributos	Descripción
-metodoEst:String	Es la selección del método de estimación hecha por el usuario
Métodos	Descripción
+reeenviarPagina()	Según el método de la Estimación, se envía a las páginas correspondientes.

Tabla XIX. Atributos y métodos de la clase en el servidor *ProcesarRecoleccion*.

Clase: <i>ProcesarRecoleccion</i> <<servidor>>	
Descripción: página del lado del servidor que administra la información de las métricas en el servidor de la base de datos.	
Atributos	Descripción
-valormin:String	Corresponde al valor mínimo de la estimación.
-valormax:String	Corresponde al valor máximo de la estimación.
-estrategia:String	Corresponde a la estrategia seleccionada
-racional:String	Es el campo que especifica el proceso de la Estimación.
-suposiciones:String	Son las suposiciones que giran alrededor de la Estimación.
-paginaActual:String	Es la pagina hacia donde se redirigirá el flujo de la aplicación.
-accion:String	Corresponde a una guía que busca controlar la navegación.
Métodos	Descripción
+registrarValorActual()	Método para registrar el valor actual en la base de datos.
+reenviarPagina()	Método que controla el flujo de la navegación.
-idMetrica:int	Es la métrica seleccionada según los elementos de la IS.

Tabla XX. Atributos y métodos de la clase en el servidor *Modelos*.

Clase: <i>Modelos</i> <<servidor>>	
Descripción: Página del lado del servidor que obtienes los modelos según la métrica, así como los parámetros críticos del modelo.	
Atributos	Descripción
Métodos	Descripción
+obtModelos()	Método para obtener los modelos según el identificador de la métrica.
+obtParametros(idModelo:int)	Método para obtener los parámetros críticos según el modelo.

Tabla XXI. Atributos y métodos de la clase en el cliente *DefinirModelo*.

Clase: <i>DefinirModelo</i> <<pagina>>	
Descripción: Página para generar la definición de un modelo.	
Atributos	Descripción
-modelos:combo	Propiedad para establecer la conexión con el servidor de base de datos.
-nombreModelo:cajaTexto	Indica la ruta y nombre de la base de datos.

Tabla XXI (cont). Atributos y métodos de la clase en el cliente DefinirModelo.

Atributos	Descripción
-nombreParametro:cajaTexto	Corresponde a la caja de texto donde se introduce el nombre del parámetro
-parámetros:lista	Listado donde se muestran los parámetros
-agregarParam:boton	Botón para agregar un elemento dentro del listado
-removerParam:boton	Botón para remover un elemento dentro del listado
-actualizarParam:boton	Botón para actualizar un elemento dentro del listado

Tabla XXII. Atributos y métodos de la clase en el servidor ProcesarModelo.

Clase: <i>ProcesarModelo</i> <<servidor>>	
Descripción: página del lado del servidor que registra los valores de los parámetros de los modelos.	
Atributos	Descripción
-nombreModelo:String[]	Corresponde a los nombres de los modelos separados por identificadores.
-parámetros:String[]	Se refiere a todos los parámetros asociados a cada parámetro.
Métodos	Descripción
+insertarDatos()	Método para registrar los datos de los parámetros y de los modelos en la base de datos.

Tabla XXIII. Atributos y métodos de la clase en el servidor VerEstimaciones.

Clase: <i>VerEstimaciones</i> <<servidor>>	
Descripción: página del lado del servidor que consulta un listado de estimaciones según la métrica del elemento IS.	
Atributos	Descripción
-idElementoIS:int	Puede ser: proyecto, proceso, producto y recursos.
-identidades:int	Puede ser análisis, diseño, implementación y pruebas.
Métodos	Descripción
-generarEstimaciones()	Según la entidad y el elemento de la IS, se general las estimaciones correspondientes.

Tabla XXIV. Atributos y métodos de la clase en el cliente ReestimacionMonitoreo.

Clase: <i>ReestimacionMonitoreo</i> <<pagina>>	
Descripción: interfaz grafica para la programación de las reestimaciones y la revisión de los monitoreos.	
Atributos	Descripción
-opcProgResstimacion:multiseleccion	Botones de multi selección para programar las reestimaciones.

Tabla XXIV (cont). Atributos y métodos de la clase en el cliente ReestimacionMonitoreo.

Atributos	Descripción
<i>-opcProgValorActual: multiseleccion</i>	Botones de multi selección para programar el valor actual.
<i>-opcProgMonitoreos: multiseleccion</i>	Botones de multi selección para programar los monitoreos.
<i>-opcProgRevisionesM: multiseleccion</i>	Botones de multi selección para programar las revisiones de los monitoreos.
<i>-metricasAsoc:combo</i>	Es un combo donde aparecen las métricas asociadas.
<i>-opcProgReestimacionMetAsoc: multiseleccion</i>	Botones de multi selección para programar las reestimaciones de las métricas asociadas.
<i>-opcProgValorActualMetAsoc: multiseleccion</i>	Botones de multi selección para programar el valor actual. de las métricas asociadas
<i>-opcProgMonitoreosMetMetAsoc: multiseleccion</i>	Botones de multi selección para programar los monitoreos de las métricas asociadas
<i>-opcProgRevisionesMetAsoc:multiseleccion</i>	Botones de multi selección para programar las revisiones de los monitoreos. de las métricas asociadas

Tabla XXV. Atributos y métodos de la clase en el servidor ObtenerEventosReestimacionMonitoreos.

Clase: <i>ObtenerEventosReestimacionMonitoreos</i> <<servidor>>	
Descripción: página del lado del servidor que obtiene los datos de las reestimaciones.	
Atributos	Descripción
<i>-metodosEstimacion:Vector</i>	Método correspondiente dentro de los métodos de Estimación.
<i>-metricasAsocaidas:Vector</i>	Conjunto de mediciones que mantienen una relación con la métrica actual.
<i>-monitoreosAsociados:Vector</i>	Son los monitoreos de las métricas asociadas.

Tabla XXVI. Atributos y métodos de la clase en el servidor *ProcesarReestimacionMonitoreo*.

Clase: <i>ProcesarReestimacionMonitoreo</i> <<servidor>>	
Descripción: página del lado del servidor que procesa los datos de las reestimaciones y los monitoreos.	
Atributos	Descripción
<i>-datosMetActual: String</i>	Corresponde a la información del valor actual
<i>-datosMetAsociados: String</i>	Es la información de las métricas asociadas
Métodos	
<i>+registrarActividades()</i>	Registrar información que proviene del valor actual
<i>+recodificarDatos()</i>	Separa los datos que integran el valor actual

Tabla XXVII. Atributos y métodos de la clase en el servidor *ObtenerListadoEstimaciones*.

Clase: <i>ObtenerListadoEstimaciones</i> <<servidor>>	
Descripción: página del lado del servidor que obtiene un listado de estimaciones de acuerdo a la métrica, el elemento y entidad de la IS.	
Atributos	Descripción
<i>-idMétrica:int</i>	Se refiere a la métrica según el elemento de la IS
<i>-idElementoIS:int</i>	Puede ser: proyecto, proceso, producto y recursos.
<i>-identidades:intt</i>	Puede ser análisis, diseño, implementación y pruebas
Métodos	Descripción
<i>+obtenerEstimaciones()</i>	Método que consulta las estimaciones según idMétrica, idElementoIS y identidades

Tabla XXVIII. Atributos y métodos de la clase en el cliente *ListadoEstimaciones*.

Clase: <i>ListadoEstimaciones</i> <<pagina>>	
Descripción: página que muestra una tabla HTML con el listado de las estimaciones.	
Atributos	Descripción
<i>-listadoEst:tablaHTML</i>	Propiedad que se refiere a una tabla HTML donde se muestra el listado.
<i>-listado:multiseleccion</i>	Propiedad que indica las cajas de selección para cada elemento de la lista.

Tabla XXIX. Atributos y métodos de la clase en el cliente *EstValAct*.

Clase: <i>EstValAct</i> <<pagina>>	
Descripción: interfaz grafica para capturar el valor actual.	
Atributos	Descripción
<i>-valorActual:cajaTexto</i>	Es una caja de texto donde simplemente se captura el valor actual que se obtuvo.
<i>-descripción:areaTexto</i>	Esta área de texto sirve para registrar los comentarios referentes al valor actual

Tabla XXX. Atributos y métodos de la clase en el servidor *ObtenerRefMetricas*.

Clase: <i>ObtenerRefMetricas</i> <<servidor>>	
Descripción: página del lado del servidor que obtiene las métricas que están vigentes según la etapa de formulación.	
Atributos	Descripción
<i>-entidadesAsociadas:Vector</i>	Conjunto de entidades que mantiene una relación según las base de la Estimación
<i>-metricasAsociadas:Vector</i>	Conjunto de mediciones que mantienen una relación con la métrica actual.
<i>-idBaseEst:int</i>	Es la referencia base de la estimación.
Métodos	Descripción
<i>+obtenerResponsables()</i>	Método para obtener los responsables de los elementos asociados.
<i>+verificarImpactoMedicion()</i>	Método para consultar la existencia de un impacto actual.

A.2.5 Discusiones grupales

Para las discusiones grupales intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen en la Tabla XXXI y la Tabla XXXII.

Tabla XXXI. Atributos y métodos de la clase en el cliente *WideBand*

Clase: <i>WideBand</i> <<pagina>>	
Descripción: interfaz grafica para la convocatoria de participantes para las discusiones grupales.	
Atributos	Descripción
<i>-participantes:cajaTexto</i>	Son dos listados, uno donde se tienen los ingenieros disponibles y otro listado donde están los participantes que se han asignado.
<i>-fechaInicio:cajaTexto</i>	FechaInicio son tres listados, uno para el día, otro para mes y otro para el año, así como combos para la hora.
<i>-fechaFin:cajaTexto</i>	FechaFin son tres listados, uno para el día, otro para mes y otro para el año, así como combos para la hora.
<i>-faseProcMed:cajaOculto</i>	Corresponde a la fase del proceso de medición donde se presentan las necesidades de discusión.
Métodos	Descripción
<i>+convocarDiscusionGrupal()</i>	Corresponde al botón se envían los datos al servidor.

Tabla XXXII. Atributos y métodos de la clase en el servidor WideBand.

Clase: <i>ProcesarWideBand</i> <<servidor>>	
Descripción: página del lado del servidor para registrar los datos de la convocatoria grupal de la discusión.	
Atributos	Descripción
<i>-participantes:String</i>	Son los datos de los ingenieros enviados en una cadena de texto.
<i>-fechaInicio:Date</i>	FechaInicio es la fecha cuando el foro de discusión se abre.
<i>-fechaFin:Date</i>	FechaFin es la fecha cuando el foro de discusión se cierra.
Métodos	Descripción
<i>+crearForo()</i>	Método que crea el foro con los parámetros correspondientes.

A.2.6 Elementos genéricos interfaz gráfica

El siguiente elemento sirve como mecanismo de navegación en SARE, los atributos y métodos de dicha clase se describen en la Tabla XXXIII.

Tabla XXXIII. Atributos y métodos de la clase en el servidor ObtenerPlantillaElementosIS.

Clase: <i>ObtenerPlantillaElementosIS</i> <<servidor>>	
Descripción: página del lado del servidor para obtener los datos de los elementos de la IS.	
Atributos	Descripción
<i>-idElementoIS:int</i>	Puede ser: proyecto, proceso, producto y recursos.
<i>-idEntidadIS:int</i>	Puede ser análisis, diseño, implementación y pruebas
Métodos	Descripción
<i>+crearLigas()</i>	Corresponde a generar los hipervínculos para cada elemento de la IS

A.2.7 Monitoreos, interpretación y retroalimentación

Para el módulo de revisión de monitoreos, interpretación y retroalimentación intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen de la Tabla XXXIV a la Tabla XLV.

Tabla XXXIV. Atributos y métodos de la clase en el servidor *ObtenerListadoMonitoreos*.

Clase: <i>ObtenerListadoMonitoreos</i> <<servidor>>	
Descripción: página del lado servidor para obtener el listado de monitoreos.	
Atributos	Descripción
<i>-idElementoIS:int</i>	Puede ser: proyecto, proceso, producto y recursos.
<i>-identidades:int</i>	Puede ser análisis, diseño, implementación y pruebas
Métodos	Descripción
<i>+obtenerMonitoreos(idElementoIS:int, idEntidades:int)</i>	Obtiene el listado de monitoreos según el elemento y la entidad de la IS.

Tabla XXXV. Atributos y métodos de la clase en el cliente *ListadoMonitoreos*.

Clase: <i>ListadoMonitoreos</i> <<pagina>>	
Descripción: interfaz que genera tabla HTML con el listado de monitoreos.	
Atributos	Descripción
<i>-listado:tablaHTML</i>	Propiedad que se refiere a una tabla HTML donde se muestra el listado de los monitoreos.

Tabla XXXVI. Atributos y métodos de la clase en el cliente *DetMonitoreoGen*.

Clase: <i>DetMonitoreoGen</i> <<pagina>>	
Descripción: plantilla genérica de los datos del monitoreo.	
Atributos	Descripción
<i>-gráfica:imagen</i>	Es la imagen JPEG o GIF que representa a la gráfica.
<i>-monitoreosAsoc:boton</i>	Es el botón que abre la ventana de los monitoreos asociados.
<i>-tipoGrafica:combo</i>	Es un combo donde se tiene el tipo de representación para la gráfica
<i>-periodos:combo</i>	Es un combo donde se despliega la frecuencia que se muestra en la gráfica
<i>-nombre:cajaTexto</i>	Es una caja de texto y corresponde al nombre de la métrica
<i>-objetivo:combo</i>	Los tipos de objetivo se presentan en un listado
<i>-comentarios:cajaTexto</i>	Corresponden a comentarios en general sobre el monitoreo.
<i>-anomalias:combo</i>	Son aquellos valores que se salen de lo estimado y estos se presentan en un listado.
<i>-anomaliasHist:cajaElementos</i>	En las cajaElementos se presentan las anomalías que ya se han presentado anteriormente.
<i>-nuevaAnom:boton</i>	Con este botón se puede agregar una nueva anomalía.

Tabla XXXVI (cont). Atributos y métodos de la clase en el cliente DetMonitoreoGen.

Atributos	Descripción
<i>-infHist:boton</i>	Con este botón se accede a la información histórica presenta todos aquellos proyectos donde se han evidenciados dicha anomalía.
<i>-convocarDistGrupal:boton</i>	Con este botón se accesa a la interfaz para convocar a los ingenieros de software a una discusión.
<i>-interpretaciones:combo</i>	Es el listado de interpretaciones y esta en función de una anomalía.
<i>-infAdicional:combo</i>	En un listado se presenta información adicional que se usa para hacer mas o menos favorable las interpretaciones.
<i>-nuevaInfoAd:boton</i>	Con este botón, se pueden dar de alta nueva información adicional.
<i>-nuevaCircunstancia:boton</i>	Con este botón se agregan nuevas causas.
<i>-recList:cajaElementos</i>	El listado de recomendaciones contiene las recomendaciones para el tipo de gráfica.
<i>-nuevaRecomendacion:boton</i>	Con este botón se pueden agregar nuevas recomendaciones.
<i>-interpHist:cajaElementos</i>	Con este botón se pueden ver el comportamiento de la Estimación en proyectos históricos.
<i>-infoAdicional:cajaElementos</i>	Se presenta la información adicional en los cajaElementos y con esto se puede asociar a la circunstancia.
<i>-circunstancias:combo</i>	Se presentan en un listado las circunstancias disponibles y puede asociarse a una anomalía en particular.
<i>-acciones:cajaElementos</i>	Se presentan en un listado las acciones de retroalimentación y puede asociarse a una interpretación en particular.
<i>-nuevaAccion:boton</i>	Con este botón se accesa la pantalla para dar de alta nuevas acciones de retroalimentación.
<i>-metricasamonit:boton</i>	Con este botón se preparan las métricas que se desea monitorear para el próximo ciclo de desarrollo, así como las acciones de retroalimentación correspondientes.
<i>-rutaRetroalimentacion:boton</i>	Este botón sirve para accesar a la interfaz de la ruta de la retroalimentación.
<i>-registrarInfo:boton</i>	Con este botón se registra toda la información asociada al monitoreo.

Tabla XXXVI (cont). Atributos y métodos de la clase en el cliente DetMonitoreoGen.

Métodos	Descripción
+ <i>abrirRutaRetroali()</i>	Es la función para abrir la ruta de la retroalimentación.
+ <i>abrirMetMonit()</i>	Con esta función se abre una ventana que contiene las métricas a monitorear.
+ <i>abrirNuevoDato(idTipo:int)</i>	Esta es una función genérica la cual permite dar de alta diferentes entidades de información, tal como, anomalías, interpretaciones, acciones de retroalimentación, entre otras.

Tabla XXXVII. Atributos y métodos de la clase en el servidor DetallesMonitoreo.

Clase: <i>DetallesMonitoreo</i> <<servidor>>	
Descripción: recupera de la base de datos la información de los monitoreos.	
Atributos	Descripción
- <i>idMonitoreo:int</i>	Es el identificador que se usa para obtener los detalles del monitoreo.
Métodos	Descripción
+ <i>traerInterpretacionesHistoricas()</i>	Con esta función se obtiene el comportamiento de las interpretaciones en proyectos anteriores.
+ <i>abrirMonitoreosAsociados()</i>	Sirve para verificar los monitoreos que están asociados, el comportamiento que se tiene.

Tabla XXXVIII. Atributos y métodos de la clase en el servidor ProcesarDetallesMonitoreo.

Clase: <i>ProcesarDetallesMonitoreo</i> <<servidor>>	
Descripción: página del lado del servidor registra los datos de los monitoreos en la base de datos.	
Atributos	Descripción
- <i>idMonitoreo:int</i>	Es el identificador que se usa para obtener los detalles del monitoreo.
- <i>datosProcMed: String</i>	Son los datos que se recolectan de la pagina procesar detalles monitoreo
Métodos	Descripción
+ <i>decodificarDatos()</i>	Esta función se usa para decodificar la información que se envía a través de la pagina detalles monitoreo.

Tabla XXXIX. Atributos y métodos de la clase en el cliente *MonitoreosAsoc*.

Clase: <i>MonitoreosAsoc</i> <<pagina>>	
Descripción: interfaz grafica para navegar entre los monitoreos de las métricas vigentes.	
Atributos	Descripción
- <i>opcionesOcultar:String</i>	Presenta una configuración de cual es la información que se oculta para determinada interfaz gráfica.

Tabla XL. Atributos y métodos de la clase en el servidor *RutaRetroalimentacion*.

Clase: <i>RutaRetroalimentacion</i> <<servidor>>	
Descripción: página del lado del servidor que obtiene las decisiones que se han tomado a través del proceso de medición.	
Atributos	Descripción
- <i>idMonitoreo:int</i>	Es el identificador que se usa para obtener los detalles del monitoreo.
Métodos	Descripción
+ <i>obtenerDatos()</i>	Este método obtiene información de todo el proceso de medición y genera la ruta de retroalimentación.

Tabla XLI. Atributos y métodos de la clase en el cliente *RutaRetroalimentacion*.

Clase: <i>RutaRetroalimentacion</i> <<pagina>>	
Descripción: interfaz grafica que muestra una tabla HTML con las decisiones del proceso de medición.	
Atributos	Descripción
- <i>decisPresProcMed:tablaHTML</i>	La tabla HTML incluye las decisiones dentro del proceso de medición.
- <i>resultadoFinal:cajaTexto</i>	Caja de texto donde se registra el resultado final en la ruta de la retroalimentación.

Tabla XLII. Atributos y métodos de la clase en el servidor *MetricasAMonitorear*.

Clase: <i>MetricasAMonitorear</i> <<servidor>>	
Descripción: página del lado del servidor que obtiene las métricas que se registran para monitorear las acciones de retroalimentación.	
Métodos	Descripción
- <i>obtenerMediciones()</i>	Método para conectarse a la BD, regresa un entero, un 1 para indicar si la conexión es exitosa y 0 en caso contrario.
- <i>obtenerAccionesInterp()</i>	Método para realizar las consultas a la BD, regresa un objeto de tipo ResultSet.

Tabla XLIII. Atributos y métodos de la clase en el cliente *MetricasAMonitorear*.

Clase: <i>MetricasAMonitorear</i> <<pagina>>	
Descripción: interfaz grafica donde se asignan métricas y acciones de retroalimentación.	
Atributos	Descripción
<i>-métricas:cajaElementos</i>	Corresponde a las métricas que se espera que se vean impactadas por las acciones de retroalimentación.
<i>-accionesRetro:cajaElementos</i>	Son las acciones de retroalimentación que serán aplicadas para el siguiente ciclo de desarrollo.
<i>-enviar:boton</i>	Botón para enviar la información al servidor.

Tabla XLIV. Atributos y métodos de la clase en el cliente *AgregarNuevoElemento*.

Clase: <i>AgregarNuevoElemento</i> <<pagina>>	
Descripción: interfaz grafica para registrar anomalías, circunstancias, recomendaciones y acciones de retroalimentación.	
Atributos	Descripción
<i>-tipoElemento:String</i>	Corresponde al tipo de elemento (anomalía, circunstancia, información adicional, recomendación o acción de retroalimentación).
<i>-nombreElem:cajaTexto</i>	Corresponde al identificador del elemento.
<i>-referencia:cajaTexto</i>	La caja de texto es para la descripción de la referencia.
<i>-tipoReferencia:combo</i>	Corresponde al tipo de referencia.
<i>-descripcionRef:areaTexto</i>	Es la fuente de la referencia.

Tabla XLV. Atributos y métodos de la clase en el servidor *ProcesarNvoElemento*.

Clase: <i>ProcesarNvoElemento</i> <<servidor>>	
Descripción: página del servidor donde se registran los datos de un nuevo elemento.	
Métodos	Descripción
<i>+registrarDatosElemento()</i>	Método para recolectar la información y la registra en la base de datos.

A.2.8 Componentes genéricos

Muchas pantallas de SARE usan el componente genérico *cajaElementos*, los atributos y métodos se describen en la Tabla XLVI.

Tabla XLVI. Atributos y métodos de la clase en el cliente *cajaElementos*.

Clase: <i>cajaElementos</i> <<pagina>>	
Descripción: componente que muestra los elementos en una lista HTML.	
Atributos	Descripción
<i>-caja1:listaMultiSel</i>	Lista HTML que contiene los elementos a ser transferidos
<i>-caja2:listaMultiSel</i>	Lista HTML que contiene los elementos que ya fueron transferidos.
<i>-botonizq:boton</i>	Botón HTML para transferir elementos de la caja derecha a la izquierda
<i>-botonder:boton</i>	Botón HTML para transferir elementos de la caja izquierda a la derecha
Métodos	Descripción
<i>+moverElementos()</i>	Función para mover elementos entre las cajas.

A.2.9 Contexto administrativo

El contexto administrativo se encarga del control del proyecto, de los equipos de trabajo, requerimientos, módulos y casos de prueba. A continuación se describen las clases que intervienen en cada uno de estos módulos.

A.2.9.1 Administración del proyecto

Para el módulo de la Administración del proyecto intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen en la Tabla XLVII y la Tabla XLVIII.

Tabla XLVII. Atributos y métodos de la clase en el cliente *AdministrarProyecto*.

Clase: <i>AdministrarProyecto</i> <<pagina>>	
Descripción: página del lado del servidor que contiene los datos del proyecto.	
Atributos	Descripción
<i>-nombre:cajaTexto</i>	Caja de texto con el nombre del proyecto.
<i>-descripción:areaTexto</i>	Caja de texto con la descripción del proyecto.
<i>-tiposPro:combo</i>	Es un combo que contiene el tipo del proyecto.
<i>-subtipo:combo</i>	Es un combo que contiene el subtipo del proyecto.
<i>-costo:cajaTexto</i>	Es una caja de texto donde se introduce el capital del que se dispondrá en el proyecto
<i>-duracion:cajaTexto</i>	Es una caja de texto para introducir en meses la duración del proyecto.
<i>-esfuerzo:cajaTexto</i>	Es una caja de texto para introducir en personas-mes la limitante de los recursos humanos del proyecto.

Tabla XLVII (cont). Atributos y métodos de la clase en el cliente AdministrarProyecto.

Atributos	Descripción
-cajaCiclos:listaMultiSel	Es una lista de multi selección donde presentan los ciclos del proyecto.
-agregarCiclos:boton	Botón para agregar ciclo.
-borrarCiclo:boton	Botón para borrar ciclo.
-enviar:boton	Botón de envío de datos al servidor
Métodos	Descripción
+abrirComponenteCalendario()	Es un calendario en JavaScript, donde se puede seleccionar cualquier fecha.

Tabla XLVIII. Atributos y métodos de la clase en el servidor ProcesarAdministracion.

Clase: <i>ProcesadorAdministracion</i> <<servidor>>	
Descripción: página del lado del servidor que procesa los datos de la administración.	
Atributos	Descripción
-tipo: String	Es el tipo del proyecto
-subtipo: String	Subtipo del proyecto
-costo: String	Capital del proyecto
-duración: String	Duración (meses) que durara el proyecto
-esfuerzo: String	Personas-mes que se pueden cubrir en el proyecto
-datosCiclos: String	Cadena que contiene la información de los ciclos, así como el inicio y fin de estos.
Métodos	Descripción
+decodificarCiclos()	Método para decodificar la información para cada ciclo.
+insertarDatosAdmon()	Método para actualizar o ingresar un proyecto a las base de datos.
+insertarCiclos()	Método para actualizar o registrar los ciclos.

A.2.9.2 Equipos de trabajo

Para el módulo de los Equipos de trabajo intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen en la Tabla XLIX y la Tabla L.

Tabla XLIX. Atributos y métodos de la clase en el cliente EquiposDeTrabajo.

Clase: <i>EquiposDeTrabajo</i> <<pagina>>	
Descripción: interfaz grafica para la configuración de ingenieros de software, roles y equipos de trabajo.	
Atributos	Descripción
-liderEquipo:combo	Se presentan todos los integrantes del equipo y se puede seleccionar ahí al líder.

Tabla XLIX (cont). Atributos y métodos de la clase en el cliente EquiposDeTrabajo.

Atributos	Descripción
<i>-equipos:combo</i>	Se presentan todos los equipos.
<i>-integrantesEquipos:cajaElementos</i>	Sirve para transferir los integrantes del proyecto a un equipo en particular.
<i>-roles:multiSeleccion</i>	Son un arreglo de botones de multi selección donde se presentan los posibles roles para cada ingeniero.
<i>-asignarIngProyectos:boton</i>	Es el botón que envía al servidor la configuración de los equipos de trabajo.
<i>-nombreEquipo:cajaTexto</i>	Es una caja de texto donde se registra el nombre del equipo.
<i>-descripEquipo:cajaTexto</i>	Es un área de texto donde se da una breve descripción del equipo.
<i>-crearEquipo:boton</i>	Es un botón que crea un equipo dentro de la misma pantalla de equipos de trabajo.
<i>-nombre:cajaTexto</i>	Es una caja de texto para introducir el nombre del ingeniero.
<i>-correo:cajaTexto</i>	Es una caja de texto para introducir el correo del ingeniero.
<i>-nick:cajaTexto</i>	Es una caja de texto para introducir el nick del ingeniero.
<i>-crearIngeniero:boton</i>	Es un botón que crea un ingeniero dentro de la misma pantalla de equipos de trabajo.

Tabla L. Atributos y métodos de la clase en el servidor ProcesarEquiposDeTrabajo.

Clase: <i>ProcesarEquiposDeTrabajo</i> <<servidor>>	
Descripción: página del lado del servidor que almacena los datos del equipo de trabajo.	
Atributos	Descripción
<i>-liderDatosIngenieros:String</i>	Son los datos que contienen la información del equipo de trabajo.
Métodos	Descripción
<i>insertarDatos()</i>	Método para insertar los datos del equipo de trabajo.

A.2.9.3 Administración de requerimientos

Para el módulo de la Administración de requerimientos intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen de la Tabla LI a la Tabla LIV.

Tabla LI. Atributos y métodos de la clase en el cliente Requerimiento.

Clase: <i>Requerimiento</i> <<pagina>>	
Descripción: interfaz grafica que muestra el detalle de los requerimientos.	
Atributos	Descripción
<i>-listaReq:hipervinculo</i>	Enlace al listado de los requerimientos.
<i>-listadoVersiones:tablaHTML</i>	Tabla HTML que contiene las versiones del requerimiento.
<i>-identificador:cajaTexto</i>	Caja de texto para introducir un identificador del requerimiento
<i>-nombre:cajaTexto</i>	Sirve para registrar un requerimiento.
<i>-prioridades:combo</i>	Posibles prioridades del requerimiento.
<i>-status:combo</i>	Posibles estados del requerimiento.
<i>-equipoAsignado:combo</i>	Listado de los posibles equipos que se hagan cargo del requerimiento.
<i>-iteraciones:combo</i>	Listado de en que iteración se desarrollara un requerimiento.
<i>-etapas:combo</i>	Etapas de en que etapa del desarrollo se encuentra un requerimiento
<i>-descripción:areaTexto</i>	Es un área de texto para la descripción del requerimiento.
<i>-racionalReq:areaTexto</i>	Área de texto para la descripción de la racional del requerimiento.
<i>-requerimientosAsoc[]:cajaElementos</i>	Sirve para establecer relaciones entre requerimientos.
<i>-racionalVersion:areaTexto</i>	Es un área de texto donde se introduce la racional de la versión.
<i>-actualizarReq:boton</i>	Botón que envía la información al servidor.

Tabla LII. Atributos y métodos de la clase en el cliente listadoRequerimientos.

Clase: <i>listadoRequerimientos</i> <<pagina>>	
Descripción: interfaz grafica que muestra el listado de los requerimientos.	
Atributos	Descripción
<i>-listado:tablaHTML</i>	Es una tabla HTML que contiene el listado de los requerimientos.
<i>-selección:multiseleccion</i>	Propiedad que indica las cajas de selección para cada elemento de la lista.
<i>-borrar:boton</i>	Botón para eliminar un conjunto de requerimientos
<i>-agregarReq:boton</i>	Botón para agregar un nuevo requerimiento.

Tabla LIII. Atributos y métodos de la clase en el servidor listadoRequerimientos.

Clase: listadoRequerimientos <<servidor>>	
Descripción: página del lado del servidor que obtiene los requerimientos.	
Atributos	Descripción
-ciclo:int	Es el ciclo actual al momento de acceder el sistema.
-idProyecto:int	Es el identificador del proyecto al momento de acceder el sistema.
Métodos	Descripción
+obtenerListadoReq()	Obtiene los requerimientos según el ciclo y proyecto actual.

Tabla LIV. Atributos y métodos de la clase en el servidor ProcesarRequerimiento.

Clase: ProcesarRequerimiento <<servidor>>	
Descripción: página del lado del servidor para registrar un requerimiento.	
Atributos	Descripción
-identificador:int	Identificador del requerimiento.
-nombre: String	Nombre del requerimiento.
-prioridad:int	Prioridad del requerimiento.
-status:int	Estado del requerimiento.
-equipoAsignado:int	Identificador del equipo asignado al requerimiento.
Atributos	Descripción
-iteración:int	Equipo asignado al requerimiento.
-etapa:int	Etapas del requerimiento.
-descripción: String	Descripción del requerimiento.
-racionalVersion: String	Racional de la versión del requerimiento.
-requerimientosAsoc: String	Cadena codificada para los requerimientos asociados
Métodos	Descripción
+decodificadorDatos()	Método para decodificar toda la información referente a los requerimientos asociados.
+insertarDatos()	Método para insertar los datos de un requerimiento.
+obtenerVersion()	Generar una nueva versión del requerimiento.

A.2.9.4 Administración de módulos

Para el módulo de la Administración de requerimientos intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen de la Tabla LV a la Tabla LVII.

Tabla LV. Atributos y métodos de la clase en el cliente DetallesModulos.

Clase: DetallesModulos <<pagina>	
Descripción: interfaz grafica que muestra los detalles de los módulos.	
Atributos	Descripción
-listadoModulos:hipervínculo	Es un hipervínculo al listado de los módulos
-listado versiones:tablaHTML	Es una tabla HTML que contiene el listado de las versiones del modulo.
-nombre:cajaTexto	Caja de texto para introducir un nombre del modulo.
-fases:cajaTexto	Listado de las posibles fases del modulo.
-prioridades:combo	Listado de las posibles prioridades del modulo.
-status:combo	Listado de los posibles estados del modulo.
-en trabajo:combo	Listado de los posibles equipos que se hagan cargo del modulo.
-iteraciones:combo	Listado de en que iteración se desarrollara un modulo.
-en pruebas:multiseleccion	Botón multi selección para especificar si ya esta en pruebas el modulo.
-descripción:cajaTexto	Es un área de texto para la descripción del modulo.
-racional ver:cajaTexto	Es un área de texto donde se introduce la racional de la versión.
-reqAsoc:cajaElementos	Uso de la caja elementos para establecer relaciones con requerimientos.
-subModulosAsoc:listaMultiSel	Uso de la caja elementos para crear submodulos.
-crear:boton	Botón crear para crear un submodulo.
-actualizar:boton	Botón para actualizar los módulos.
-eliminar:boton	Botón para eliminar los submodulos.
-modulosAsoc:cajaElementos	Sirve para establecer relaciones con los módulos.
-enviar:boton	Botón de enviar para transferir la información al servidor.

Tabla LVI. Atributos y métodos de la clase en el cliente listadoModulos.

Clase: listadoModulos <<pagina>>	
Descripción: interfaz grafica que muestra el listado de los módulos.	
Atributos	Descripción
-listado:tablaHTML	Es una tabla HTML que contiene el listado de los módulos.
-selección:multiseleccion	Indica las cajas de selección para cada elemento de la lista.
-borrar:boton	Botón para eliminar un conjunto de módulos
-agregarmod:boton	Botón para agregar un nuevo módulo.

Tabla LVII. Atributos y métodos de la clase en el servidor **ProcesarDatosModulos**.

Clase: <i>ProcesarDatosModulos</i> <<servidor>>	
Descripción: página del lado del servidor que procesa los datos de los módulos.	
Atributos	Descripción
<i>-nombre:cajaTexto</i>	Nombre del módulo.
<i>-fase:cajaTexto</i>	Fase del módulo.
<i>-prioridad:combo</i>	Prioridad del módulo
<i>-status:combo</i>	Estado del módulo
<i>-eqtrabajo:combo</i>	Equipo de trabajo asignado al módulo.
<i>-iteración:combo</i>	Iteración del módulo.
<i>-enpruebas:multiseleccion</i>	Bandera para saber si el módulo esta o no en la etapa de pruebas
<i>-descripcion: String</i>	Descripción del módulo.
<i>-racionalVersion: String</i>	Racional de la versión del módulo.
<i>-reqAsoc: String</i>	Cadena codificada para los requerimientos asociados
<i>-modAsoc: String</i>	Cadena codificada para los módulos asociados
<i>-submodAsoc: String</i>	Cadena codificada para los submodulos asociados.
Métodos	Descripción
<i>decodificarDatos()</i>	Método para decodificar toda la información referente a los módulos asociados.
<i>insertaDatos()</i>	Método para insertar los datos de un módulo.
<i>obtenerVersion()</i>	Método para generar una nueva versión del módulo.

A.2.9.5 Administración de casos de prueba

Para el módulo de la Administración de requerimientos intervienen las siguientes páginas web que mantienen un comportamiento tanto en el cliente como en el servidor, los atributos y métodos de dichas clases se describen de la Tabla LVIII a la Tabla LXI.

Tabla LVIII. Atributos y métodos de la clase en el servidor **ListadoPruebas**.

Clase: <i>ListadoPruebas</i> <<servidor>>	
Descripción: página del lado del servidor obtiene los casos de prueba.	
Atributos	Descripción
<i>-ciclo:int</i>	Es el ciclo actual al momento de acceder el sistema.
<i>-idProyecto:int</i>	Identificador del proyecto al momento de acceder el sistema.
Métodos	Descripción
<i>+obtenerCasosPrueba()</i>	Obtiene los casos de prueba según el ciclo y proyecto actual.

Tabla LIX. Atributos y métodos de la clase en el cliente ListadoPruebas.

Clase: ListadoPruebas <<pagina>>	
Descripción: interfaz grafica que muestra el listado de las pruebas.	
Atributos	Descripción
-listado:tablaHTML	Es una tabla HTML que contiene el listado de los casos de prueba.
-selección:multiseleccion	Propiedad que indica las cajas de selección para cada elemento de la lista.
-borrar:boton	Botón para eliminar un conjunto de casos de prueba
-agregarPrueba:boton	Botón para agregar un nuevo módulo.

Tabla LX. Atributos y métodos de la clase en el cliente DetallesPruebas.

Clase: DetallesPruebas <<pagina>>	
Descripción: interfaz grafica que muestra el detalle de las pruebas.	
Atributos	Descripción
-listadoVersiones:tablaHTML	Es una tabla HTML que contiene el listado de las versiones de los casos de prueba.
-nombre:cajaTexto	Caja de texto para introducir un nombre del caso de prueba.
-descripción:areaTexto	Descripción del caso de prueba.
-tipodePruebas:combo	Listado de los posibles tipos de prueba.
-elementoISAsociado:cajaElementos	Listado de los posibles elementos de la IS asociado.
-racionalVersion:areaTexto	Racional de la versión del caso de prueba.
-status:combo	Estado del casos de prueba
-equipo:combo	Equipo de trabajo asignado a los casos de prueba.
-iteraciones:combo	Iteración de los casos de prueba.

Tabla LXI. Atributos y métodos de la clase en el servidor ProcesarPruebas.

Clase: ProcesarPruebas <<Servidor>>	
Descripción: página del lado del servidor para procesar la información de las pruebas.	
Atributos	Descripción
-nombre: String	Nombre del módulo.
-descripción: String	Descripción del módulo.
-racionalVersion: String	Racional de la versión del módulo
-tipodePrueba:int	Identificador del tipo de prueba
-status:int	Estado del módulo
-equipo:int	Equipo de trabajo asignado al módulo.
-iteración:int	Iteración del módulo
-elementosAsociados: String	Cadena codificada para los elementos asociados.

Tabla LXI (cont). Atributos y métodos de la clase en el servidor ProcesarPruebas.

Métodos	Descripción
<i>+decodificaElementos()</i>	Método para decodificar toda la información referente a los módulos asociados.
<i>+registrarDatos()</i>	Método para insertar los datos de un módulo.
<i>+obtenerVersion()</i>	Método para generar una nueva versión del módulo.

A.3 Modelo de datos (tablas).

El modelo de datos corresponde al diseño de las tablas que se hizo bajo el esquema de entidad-relación para darle soporte a los datos que se requieren almacenar en el sistema SARE.

A.3.1 Contexto administrativo

A continuación se muestran las tablas correspondientes a la administración del proyecto, los equipos de trabajo, la administración de requerimientos, módulos y casos de prueba.

A.3.1.1 Administración del proyecto

La estructura de las tablas que intervienen para la administración del proyecto se presenta de la Tabla LXII a la Tabla LXIV.

Tabla LXII. Estructura de la tabla Proyectos.

Tabla	Proyectos	
Descripción: tabla que almacena la información de los proyectos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del proyecto.
nombre	varchar(20)	Nombre del proyecto.
liderad	int(11)	Identificador del líder del proyecto.
tipo	int(11)	Tipo de proyecto.
subtipo	int(11)	Subtipo de proyecto.
descripcion	varchar(150)	Descripción del proyecto.
fechaInicio	date	Fecha de inicio del proyecto.
fechaFin	date	Fecha fin del proyecto.
presupuesto	varchar(12)	Presupuesto del proyecto.
maxEsfuerzo	int(11)	Máximo esfuerzo del proyecto.
moneda	char(1)	Identificador de moneda para el proyecto.

Tabla LXIII. Estructura de la tabla tipoProy.

Tabla	tipoProy	
Descripción: tabla que almacena los tipos de proyecto.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un tipo de proyecto.
nombre	varchar(80)	Nombre del tipo de proyecto.

Tabla LXIV. Estructura de la tabla ciclos.

Tabla	Ciclos	
Descripción: tabla que almacena la información de los ciclos asociados a los proyectos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del ciclo.
descripcion	varchar(60)	Descripción del ciclo.
fechaInicio	date	Fecha de inicio del ciclo.
fechaFin	date	Fecha fin del ciclo.
idProy	int(11)	Identificador del proyectos asociado al ciclo.

A.3.1.2 Administración de equipos de trabajo

La estructura de las tablas que intervienen para la administración de los equipos de trabajo se presenta de la Tabla LXV a la Tabla LXVIII.

Tabla LXV. Estructura de la tabla Equipos.

Tabla	Equipos	
Descripción: Tabla que almacena la información de los equipos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un equipo.
nombre	varchar(60)	Nombre del equipo.
descripcion	varchar(150)	Descripción del equipo.
id_ciclo	int(11)	Identificador del ciclo asociado.
idLider	int(11)	Identificador del líder asociado
id_proy	int(11)	Identificador del proyecto asociado.

Tabla LXVI. Estructura de la tabla Usuarios_eq_rol.

Tabla	Usuarios_eq_rol	
Descripción: Almacena la relación de equipos, roles asociados a proyectos y usuarios.		
Nombre del campo	Tipo de dato	Descripción
id_usuario	int(11)	Identificador de un usuario asociado a un rol.
id_equipo	int(11)	Identificador del equipo asociado a un rol.
rol	char(1)	Identificador de un rol.
id_proy	int(11)	Proyecto asociado a un(os) rol(es).

Tabla LXVII. Estructura de la tabla Usuarios.

Tabla	Usuarios	
Descripción: tabla que almacena la información de los usuarios.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un usuario.
nombre	varchar(60)	Nombre del usuario.
passwd	varchar(8)	Contraseña del usuario.
nick	varchar(60)	Sobrenombre de usuario.
email	varchar(60)	Correo electrónico del usuario.

Tabla LXVIII. Estructura de la tabla usuariosNivelesLideresProy.

Tabla	usuariosNivelesLideresProy	
Descripción: tabla que almacena la relación de usuarios con equipos y proyectos.		
Nombre del campo	Tipo de dato	Descripción
id_usuario	int(11)	Identificador del usuario.
nivelEq	int(11)	Nivel de equipo.
id_proy	int(11)	Identificador de proyecto.

A.3.1.3 Administración requerimientos

La estructura de las tablas que intervienen para la administración de los equipos de trabajo se presenta en la Tabla LXIX y la Tabla LXX.

Tabla LXIX. Estructura de la tabla Requerimientos.

Tabla	Requerimientos	
Descripción: tabla que almacena la información de los requerimientos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del requerimiento.
nombre	varchar(30)	Nombre del requerimiento.
iteracion	varchar(20)	Número de iteración del requerimiento.
versión	int(11)	Versión del requerimiento.
Nombre del campo	Tipo de dato	Descripción
id_equipo	int(11)	Identificador del equipo.
id_proy	int(11)	Identificador del proyecto
autor	int(11)	Identificador del usuario
fechad	date	Fecha de creación del requerimiento.
descripcion	varchar(250)	Descripción del requerimiento.
racional	varchar(250)	Racional del requerimiento.

Tabla LXX. Estructura de la tabla relacionesReq.

Tabla	relacionesReq	
Descripción: tabla que almacena las relaciones de los requerimientos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un requerimiento.
idRelacion	int(11)	Identificador de requerimiento relacionado.

A.3.1.4 Administración de Módulos

La estructura de las tablas que intervienen para la administración de los requerimientos se presenta de la Tabla LXXI a la Tabla LXXIV.

Tabla LXXI. Estructura de la tabla Modulos.

Tabla	Módulos	
Descripción: tabla que almacena los módulos asociados a los equipos y usuarios.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un módulo.
version	int(11)	Versión del módulo.
nombre	varchar(30)	Nombre del módulo.
descripcion	varchar(250)	Descripción del módulo.
racionalVersion	varchar(250)	Racional versión del módulo.
iteracion	int(11)	Número de iteración del módulo.
fase	char(1)	Identificador de la fase del módulo.
autor	int(11)	Identificador de usuario.
equipo	int(11)	Identificador del equipo.
prioridad	char(1)	Identificador del nivel de prioridad.
status	char(1)	Identificador del status del módulo.
fechaC	Date	Fecha de creación del módulo.

Tabla LXXII. Estructura de la tabla reqModulos.

Tabla	reqModulos	
Descripción: tabla que almacena la relación de módulos con requerimientos.		
Nombre del campo	Tipo de dato	Descripción
idModulo	int(11)	Identificador de un módulo.
idRequerimiento	int(11)	Identificador de un requerimiento.

Tabla LXXIII. Estructura de la tabla **modulosSubmodulos**.

Tabla	modulosSubmodulos	
Descripción: tabla que almacena la relación de submodulos con módulos.		
Nombre del campo	Tipo de dato	Descripción
idModulo	int(11)	Identificador de un módulo.
idSubModulo	int(11)	Identificador de un submodulo.

Tabla LXXIV. Estructura de la tabla **Modulos_mod**.

Tabla	Modulos_mod	
Descripción: tabla que almacena la relación de módulos asociados con los módulos.		
Nombre del campo	Tipo de dato	Descripción
idModulo	int(11)	Identificador de un módulo.
idModuloRel	int(11)	Identificador de un módulo asociado.

A.3.1.5 Tablas auxiliares en SARE

La estructura de las tablas auxiliares que intervienen para apoyar el contexto de SARE se presenta de la Tabla LXXV a la Tabla LXXVIII.

Tabla LXXV. Estructura de la tabla **procesosIS**.

Tabla	procesosIS	
Descripción: tabla que almacena la información de los procesos de la IS.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del proceso de la IS.
proceso	varchar(30)	Nombre del proceso de la IS.

Tabla LXXVI. Estructura de la tabla **productosIS**.

Tabla	productosIS	
Descripción: tabla que almacena la información de los productos de la IS.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del producto de la IS.
producto	varchar(30)	Nombre del producto de la IS.

Tabla LXXVII. Estructura de la tabla **elementosIS**.

Tabla	elementosIS	
Descripción: tabla que almacena la información de los elementos de la IS.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del elemento ortogonal de la IS.
elemento	varchar(30)	Nombre del elemento de la IS.

Tabla LXXVIII. Estructura de la tabla Fases.

Tabla	Fases	
Descripción: tabla que almacena la información de las fases.		
Nombre del campo	Tipo de dato	Descripción
fase	char(1)	Identificador de la fase.
descripcion	varchar(60)	Descripción de la fase.

A.3.2 Formulación de mediciones

La estructura de las tablas que intervienen para el módulo de la Formulación de mediciones se presenta de la Tabla LXXIX a la Tabla LXXXV.

Tabla LXXIX. Estructura de la tabla catalogoMetricasContextoIS.

Tabla	catalogoMetricasContextoIS	
Descripción: almacena las relaciones de los elementos y entidades de la IS con el catalogo de las métricas.		
Nombre del campo	Tipo de dato	Descripción
idCatalogoM	int(11)	Identificador del catálogo de métricas.
idelementoIS	int(11)	Identificador del elemento ortogonal de la IS.
identidadIS	int(11)	Identificador de la entidad de la IS.

Tabla LXXX. Estructura de la tabla catalogoMetricas.

Tabla	catalogoMetricas	
Descripción: tabla que almacena la información de los catálogos de métricas.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del catálogo de métricas.
descripcion	varchar(60)	Descripción del catálogo de métricas.

Tabla LXXXI. Estructura de la tabla configuracionActualMetricas.

Tabla	configuracionActualMetricas	
Descripción: tabla que almacena la configuración actual de las métricas.		
Nombre del campo	Tipo de dato	Descripción
idCatalogoM	int(11)	Identificador del catálogo de las métricas.
idelementoIS	int(11)	Identificador del elemento ortogonal de la IS.
identidadIS	int(11)	Identificador de la entidad de la IS.
idCiclo	int(11)	Identificador de un ciclo.

Tabla LXXXII. Estructura de la tabla relacionMetricas.

Tabla	relacionMetricas	
Descripción: tabla que almacena la relación de las métricas		
Nombre del campo	Tipo de dato	Descripción
Id	int(11)	Identificador de las relaciones de métricas.
idMed	int(11)	Identificador de la medición.
idMedicionConsec	int(11)	Identificador de la medición asociada.
tipo	int(11)	Identificador del tipo de medición.

Tabla LXXXIII. Estructura de la tabla perfilesMetricas.

Tabla	perfilesMetricas	
Descripción: tabla que almacena el perfil de la métrica.		
Nombre del campo	Tipo de dato	Descripción
Id	int(11)	Identificador de un perfil de métricas.
Descripción	varchar(80)	Descripción de perfil métrica.
Unidades	varchar(45)	Unidades de las métricas.

Tabla LXXXIV. Estructura de la tabla referenciasprocMedicion.

Tabla	referenciasprocMedicion	
Descripción: tabla que almacena las bases de proceso de medición.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de referencia al proceso de medición.
idfase	int(11)	Identificador de la fase.
subFase	double	Identificador de la subfase.
nombre	varchar(80)	Nombre de la referencia al proceso de medición.

Tabla LXXXV. Estructura de la tabla recalendarizarAccionesProcMedicion.

Tabla	recalendarizarAccionesProcMedicion	
Descripción: tabla que almacena cuando se llevara a cabo una acción referente al proceso de medición.		
Nombre del campo	Tipo de dato	Descripción
idrefProcMedDet	float	Referencia del proceso de medición detallado
idElementoBaseProcMed	int(11)	Elemento base del proceso de medición.
idTipoEvento	int(11)	Tipo de evento
diasPrevios	int(11)	Días previos con que se envía una notificación.

A.3.3 Recolección

La estructura de las tablas que intervienen para el módulo de recolección se presenta de la Tabla LXXXVI a la Tabla XC.

Tabla LXXXVI. Estructura de la tabla Modelos.

Tabla	Modelos	
Descripción: tabla que registra los datos de modelo de las estimaciones.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del modelo.
idCatalogoM	int(11)	Identificador del catalogo de métricas.
nombreModelo	varchar(30)	Nombre del modelo.

Tabla LXXXVII. Estructura de la tabla parametrosModelos.

Tabla	parametrosModelos	
Descripción: tabla que almacena los parámetros de los modelos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del parámetro de un modelo.
idModelo	int(11)	Identificador del modelo.
nombreParam	varchar(60)	Nombre del parámetro.

Tabla LXXXVIII. Estructura de la tabla valorParametrosModelos.

Tabla	valorParametrosModelos	
Descripción: tabla que contiene los valores de los parámetros de los modelos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del valor de los parámetros del modelo.
idModelo	int(11)	Identificador del modelo.
idParam	varchar(60)	Identificador del parámetro.
valor	double	Valor del parámetro.

Tabla LXXXIX. Estructura de la tabla baseEstimaciones.

Tabla	baseEstimaciones	
Descripción: tabla que contiene las estimaciones originales de los elementos de la IS		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de base estimaciones.
idElementoIS	int(11)	Identificador de elemento ortogonal de la IS.
idEntidadIS	int(11)	Identificador de la entidad de la IS.
idProyecto	int(11)	Identificador de proyecto.
idCatalogoM	int(11)	Identificador del catalogo de métricas.
tipo	char(1)	Identificador del tipo de base estimación

Tabla XC. Estructura de la tabla valoresEstimaciones.

Tabla	valoresEstimaciones	
Descripción: tabla que registra las versiones de las estimaciones.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de valores de las estimaciones.
idBaseEstOriginal	int(11)	Identificador de la estimación base original.
Nombre del campo	Tipo de dato	Descripción
versión	int(11)	Versión del valor de la estimación.
idCiclo	int(11)	Identificador del ciclo.
valorMin	double	Valor mínimo de la estimación.
valorMax	double	Valor máximo de la estimación.
fecha	date	Fecha de creación del valor de la estimación.

A.3.4 Monitoreos, interpretación y retroalimentación

La estructura de las tablas que intervienen para la revisión de los monitoreos, la interpretación y la retroalimentación se presenta de la Tabla XCI a la Tabla CIV.

Tabla XCI. Estructura de la tabla cfgMonitoreosActuales.

Tabla	cfgMonitoreosActuales	
Descripción: tabla que almacena los monitoreos vigentes.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del monitoreo actual.
idTipoMonitoreo	int(11)	Identificador del tipo de monitoreo.
idRelMetricas	int(11)	Identificador de la relación de métricas.
idBaseEst	int(11)	Identificador de la base estimación.
porCapturar_Band	int(11)	Identificador del WideBand.
díasPrevios	int(11)	Número de días previos con que se envía una notificación.
fechaC	date	Fecha de la creación de la configuración del monitoreo actual.

Tabla XCII. Estructura de la tabla tipoObjMonitoreo.

Tabla	tipoObjMonitoreo	
Descripción: tabla que contiene los tipos de los objetivos de monitoreo.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del tipo objetivo monitoreo.
abrev	char(2)	Abreviación del tipo objetivo monitoreo.
descripcion	varchar(60)	Descripción del tipo objetivo monitoreo.

Tabla XCIII. Estructura de la tabla referenciaRacional.

Tabla	referenciaRacional	
Descripción: tabla que contiene la racional de la referencia.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de la referencia racional.
idRelacion	int(11)	Identificador de la relación entre métricas.
idTipoMonitoreo	int(11)	Identificador del tipo de monitoreo.
descripcion	varchar(60)	Descripción de la referencia racional.
comentarios	varchar(60)	Comentarios sobre la referencia racional.
idTipoReferencia	int(11)	Identificador del tipo de referencia.
dominio	char(1)	Es el nivel de aplicación de la referencia racional

Tabla XCIV. Estructura de la tabla racionalMonitoreos.

Tabla	racionalMonitoreos	
Descripción: tabla que contiene la racional de los monitoreos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de racional monitoreos.
idMonitoreo	int(11)	Identificador de monitoreo.
versión	int(11)	Versión de la racional monitoreos.
Nombre del campo	Tipo de dato	Descripción
idTipoObjMonitoreo	int(11)	Identificador del tipo de objeto monitoreo.
idCiclo	int(11)	Identificador del ciclo.
comentariosR	Text	Comentarios sobre racional monitoreos.
idRefRacional	int(11)	Identificador de la referencia racional.
relValida	int(11)	Identificador de la relación valida.

Tabla XCV. Estructura de la tabla tiposReferenciaRacional.

Tabla	tiposReferenciaRacional	
Descripción: tabla que contiene la racional de los tipos de referencia.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un tipo referencia racional.
tipo	char(1)	Acrónimo del tipo de referencia racional.
descripcion	varchar(60)	Descripción del tipo de referencia racional.

Tabla XCVI. Estructura de la tabla valoresAnomalos.

Tabla	valoresAnomalos	
Descripción: tabla que contiene los valores anómalos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de un valor anómalo.
titulo	varchar(60)	Título de un valor anómalo.
descripcion	text	Descripción de un valor anómalo.
idRacionalRelacion	int(11)	Identificador de la relación racional.
idTipoAnomalia	int(11)	Identificador del tipo de anomalía.

Tabla XCVII. Estructura de la tabla tiposValoresAnomalos.

Tabla	tiposValoresAnomalos	
Descripción: tabla que contiene los tipos de los valores anómalos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del tipo valor anómalo.
abrev	char(2)	Abreviación del tipo valor anómalo.
descripcion	varchar(60)	Descripción del tipo valor anómalo.

Tabla XCVIII. Estructura de la tabla catalogoMonitoreos.

Tabla	catalogoMonitoreos	
Descripción: tabla que contiene los tipos de monitoreo.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del catalogo de monitoreos.
descTipoMonitoreo	varchar(60)	Descripción del catalogo de monitoreos.

Tabla XCIX. Estructura de la tabla TipoEventos.

Tabla	TipoEventos	
Descripción: tabla que contiene los tipos de eventos.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador del tipo de evento.
tipoEvento	varchar(5)	Acrónimo del tipo de evento.
descripcion	varchar(50)	Descripción del tipo de evento.
tipo	char(2)	Es la clasificación del tipo del evento.

Tabla C. Estructura de la tabla Causas.

Tabla	Causas	
Descripción: tabla que contiene las causas de las anomalías según el tipo de monitoreo.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de la causa.
titulo	varchar(60)	Título de la causa.
descripcion	text	Descripción de la causa.
excepcion	int(11)	Id de la excepción asociada
tipoCausa	int(11)	Tipo de causa

Tabla CI. Estructura de la tabla MasInformacion.

Tabla	MasInformacion	
Descripción: tabla que contiene información adicional que apoya a las causas.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de la información adicional.
descripcion	text	Descripción de la información adicional.
idCausa	int(11)	Identificador de la causa asociada

Tabla CII. Estructura de la tabla Rel_anomalia_causa.

Tabla	Rel_anomalia_causa	
Descripción: tabla que contiene la relación de las anomalías con las causas.		
Nombre del campo	Tipo de dato	Descripción
idRelacion	int(11)	Identificador de la relación.
idAnomalia	int(11)	Identificador de la anomalía.
idCausa	int(11)	Identificador de la causa.

Tabla CIII. Estructura de la tabla AccionRetroalimentacion.

Tabla	AccionRetroalimentacion	
Descripción: tabla que contiene las acciones de la retroalimentación.		
Nombre del campo	Tipo de dato	Descripción
id	int(11)	Identificador de la acción de retroalimentación.
titulo	varchar(60)	Título de la acción de retroalimentación.
descripcion	text	Descripción de la acción de retroalimentación.

Tabla CIV. Estructura de la tabla Metricas_a_monitorear.

Tabla		Metricas_a_monitorear
Descripción: tabla que contiene la relación de las métricas con las causas y la retroalimentación,		
Nombre del campo	Tipo de dato	Descripción
idRetroalimentacion	int(11)	Identificador de la retroalimentación.
idRelMetricas	int(11)	Identificador de la relación entre métricas.
tipoCausa	int(11)	Tipo de causa

Apéndice B. Datos de las pruebas

Los casos de prueba definidos para este Apéndice, tienen como objetivo reflejar que la funcionalidad de SARE haya sido cubierta.

El escenario que se plantea corresponde a un proceso de desarrollo de software en equipos de trabajo donde participan diversos roles. Principalmente, el líder esta interesado en conocer cual es el estado del proyecto y para esto necesita conocer el desempeño de diferentes áreas, cada área cuenta con un líder y estos son los encargados de reportar información al líder.

A continuación se describen los datos que se utilizaron para cubrir estas pruebas. El orden en el que se hizo fue siguiendo como un usuario real interactuaría con SARE.

B.1 Alta del proyecto

Los datos proyecto de prueba que esta dado de alta en el sistema se describe en la Tabla CV.

Tabla CV. Datos del proyecto de prueba.

Datos del proyecto				
Proyecto ID	000001			
Nombre	Proyecto de prueba			
Fecha Inicial	05/03/06			
Fecha Final	15/09/06			
Objetivos				
Fases del proyecto	Nombre	Costo	Duración	Inicio
	Análisis	\$15,000.00	50	05/03/06
	Implementación	\$28,963.00	63	02/04/06
	Pruebas	\$40,000.00	60	15/06/06

La Tabla CVI muestra los ciclos de trabajo para el proyecto de prueba.

Tabla CVI. Ciclos de trabajo del proyecto de prueba.

Ciclos de trabajo		
Fecha Inicio	Fecha Fin	Ciclo
05/03/06	30/04/06	Ciclo 1
01/05/06	05/06/06	Ciclo 2
05/06/06	15/09/06	Ciclo 3

B.2 Roles y equipos de trabajo

Básicamente consistió en detallar ingenieros de software y equipos de trabajo que participen en un proyecto y donde cada líder tenga necesidades de información específicas. Los equipos de trabajo se describen en la Tabla CVII.

Tabla CVII. Requerimientos del proyecto de prueba.

Equipo Corporativo				
Miembro ID	Nombre	Rol	Email	Teléfono
<i>Luis</i>	<i>Luis Lobo</i>	<i>líder proyecto</i>	lobo@hotmail.com	<i>178-0000</i>
<i>Vanes</i>	<i>Vanessa Lopez</i>	<i>líder análisis</i>	vanes@gmail.com	<i>178-0000</i>
<i>Rodri</i>	<i>Rodrigo Glez</i>	<i>Líder diseño</i>	roy@sare.com	<i>178-0000</i>
<i>Fergoma</i>	<i>Fernanda Gomez</i>	<i>líder desarrollo</i>	fergoma@sols.com	<i>176-1200</i>
<i>Aliv</i>	<i>Angel Liva</i>	<i>líder pruebas</i>	aliva@dask.com	<i>176-3454</i>
Equipo Análisis				
Miembro ID	Nombre	Rol	Email	Teléfono
<i>vanes</i>	<i>Vanessa López</i>	<i>líder análisis</i>	vanes@gmail.com	<i>178-0000</i>
<i>ramiro</i>	<i>Ramiro Mendoza</i>	<i>Analista</i>	mendoz@gmail.com	<i>178-2134</i>
<i>luiszarate</i>	<i>Luis Zarate</i>	<i>Analista</i>	rluis@sare.com	<i>178-0100</i>
Equipo Diseño				
Miembro ID	Nombre	Rol	Email	Teléfono
<i>rodri</i>	<i>Rodrigo Glez</i>	<i>Líder diseño</i>	roy@sare.com	<i>178-0000</i>
<i>betty</i>	<i>Beatriz Mtz</i>	<i>Diseñador</i>	betty@sols.com	<i>176-1200</i>
Equipo Desarrollo				
Miembro ID	Nombre	Rol	Email	Teléfono
<i>fergoma</i>	<i>Fernanda Gómez</i>	<i>Líder desarrollo</i>	fergoma@sols.com	<i>176-1200</i>
<i>betty</i>	<i>Beatriz Mtz</i>	<i>Desarrollador</i>	betty@sols.com	<i>176-1210</i>
<i>carlos</i>	<i>Carlos Hdez</i>	<i>Desarrollador</i>	carl@dask.com	<i>176-3434</i>

Tabla CVII (cont). Requerimientos del proyecto de prueba.

Equipo Pruebas				
Miembro ID	Nombre	Rol	Email	Teléfono
<i>aliv</i>	<i>Angel Liva</i>	<i>Líder pruebas</i>	aliva@dask.com	176-3454
<i>jgomez</i>	<i>Jose Gomez</i>	<i>Ing. Pruebas</i>	gomez@add1.com	116-1210
<i>luiszarate</i>	<i>Luis Zarate</i>	<i>Ing. Pruebas</i>	rluis@sare.com	178-0100

B.3 Contexto administrativo

Este contexto esta en función de las tareas que realiza cada ingeniero de software, a continuación se describe estas tareas.

B.3.1 Requerimientos

Los requerimientos para el proyecto de prueba se describen en la Tabla CVIII.

Tabla CVIII. Requerimientos del proyecto de prueba.

ID	RF1
RF	Agregar, modificar, borrar tipos de contacto para el cliente
Descripción:	Agregar, modificar o eliminar los datos de los tipos de contacto del cliente.
Versión	Descripción
1.0	Se agrego la opción de activo/inactivo.
ID	RF2
RF	Agregar, modificar, borrar días festivos
Descripción:	Agregar, modificar o eliminar los días festivos.
ID	RF3
RF	Agregar, modificar, borrar grupos
Descripción:	Agregar, modificar o eliminar los datos de los grupos.
Versión	Descripción
1.0	Se agrego la opción de activo/inactivo.
ID	RF4
RF	Agregar, modificar, borrar tipos de dirección
Descripción:	Agregar, modificar o eliminar los tipos de dirección.
ID	RF5
RF	Agregar, modificar, borrar tipos de teléfono.
Descripción:	Agregar, modificar o eliminar los tipos de teléfono.

Tabla CVIII (cont). Requerimientos del proyecto de prueba.

ID	RF6
RF	Agregar, modificar, borrar tipos de contactos de las compañías.
Descripción:	Agregar, modificar o eliminar los tipos de contactos de compañías.
Versión	Descripción
1.0	Se agrego la opción de activo/inactivo.
ID	RF7
RF	Agregar, modificar, borrar usuarios
Descripción:	Agregar, modificar o eliminar los datos del usuario.
Versión	Descripción
1.0	Se agrego la opción de activo/inactivo.
2.0	Se modifíco a que el nombre sea capturado en 3 cajas de texto, uno para el nombre y otros para los apellidos.
Req asoc:	RF8
ID	RF8
RF	Agregar, modificar, borrar privilegios
Descripción:	Agregar, modificar o eliminar los datos de privilegios.
Req asoc:	RF3
ID	RF9
RF	Agregar, modificar, borrar compañías
Descripción:	Agregar, modificar o eliminar los datos de las compañías.
Versión	Descripción
1.0	Se agrego la opción de activo/inactivo.
ID	RF10
RF	Agregar, modificar, borrar contactos de compañías.
Descripción:	Agregar, modificar o eliminar los contactos de las compañías.
Versión	Descripción
1.0	Se agrego la opción de activo/inactivo.
Req asoc:	RF6
ID	RF11
RF	Agregar, modificar, borrar direcciones de la compañía.
Descripción:	Agregar, modificar o eliminar las direcciones de la compañía.
Req asoc:	RF9, RF4
ID	RF12
RF	Agregar, modificar, borrar teléfonos de la compañía.
Descripción:	Agregar, modificar o eliminar los teléfonos de la compañía.
Req asoc:	RF9, RF5

B.3.2 Diseño

Los módulos de diseño se describen en la Tabla CIX.

Tabla CIX. Módulos de diseño del proyecto de prueba.

Modulo_dis_1			
Modulo:	Usuarios y grupos	Ciclo:	Ciclo 1
Descripción:	Trata de los privilegios que tienen los usuarios y los grupos		
Prioridad:	Alta	Equipo:	Equipo 1
Req asociados:	RF3, RF7 y RF8.		
Modulo_dis_2			
Modulo:	Cat para el cliente	Ciclo:	Ciclo 2
Descripción:	Catálogos para el cliente.		
Prioridad:	Alta	Equipo:	Equipo 1
Req asociados:	RF1, RF2, RF4 y RF5.		
Modulo_dis_3			
Modulo:	Personalización de las compañías	Ciclo:	Ciclo 3
Descripción:	Definir elementos que después el cliente usaría.		
Prioridad:	Alta	Equipo:	Equipo 1
Req asociados:	RF6, RF9, RF10.		
Mod asociados:	Modulo 2.		
Submodulos:	subModulo_dis_4, subModulo_dis_5.		
subModulo_dis_4			
Modulo:	Dir de las compañías	Ciclo:	Ciclo 3
Descripción:	Definir elementos que después el cliente usaría.		
Prioridad:	Alta	Equipo:	Equipo 1
Req asociados:	RF11.		
subModulo_dis_5			
Modulo:	TEL de las compañías	Ciclo:	Ciclo 3
Descripción:	Definir elementos que después el cliente usaría.		
Prioridad:	Alta	Equipo:	Equipo 1
Req asociados:	RF12.		

B.3.3 Implementación

Los módulos de la implementación se describen en la Tabla CX.

Tabla CX. Módulos de implementación del proyecto de prueba.

Diseño	Implementación	Ciclo:
Modulo_dis_1	Modulo_imp_1	Ciclo 1
Modulo_dis_2	Modulo_imp_2	Ciclo 2
Modulo_dis_3	Modulo_imp_3	Ciclo 3

B.3.4 Casos de prueba

Para caso de prueba se sacaron 3 requerimientos. A continuación se muestran los tres casos de prueba para el primer requerimiento, estos se describen en la Tabla CXI.

Tabla CXI. Casos de prueba para el Requerimiento Funcional 1.

ID	CP1RF1
RF	Agregar días festivos.
Descripción:	Verificar que se hayan agregado días festivos en la base de datos y que se refleje esto en la interfaz gráfica.
ID	CP2RF1
RF	Modificar días festivos.
Descripción:	Verificar que se hayan modificado los días festivos en la base de datos y que se refleje esto en la interfaz gráfica.
ID	CP3RF1
RF	Eliminar los días festivos.
Descripción:	Verificar que se hayan eliminado los días festivos en la base de datos y que se refleje esto en la interfaz gráfica.

B.4 Formulación

Las métricas a monitorear se describen en la Tabla CXII.

Tabla CXII. Métricas a monitorear durante el proyecto de prueba.

Elemento IS	Entidad IS	Métrica
Proyecto	-----	Duración
Proceso	Análisis	requerimientos terminados
		Estabilidad
	Diseño	Validación módulos
	Implementación	Implementación por requerimientos
Producto	Pruebas	Casos de prueba requeridos.

B.5 Recolección

Los valores estimados para las métricas que se formularon se describen en la Tabla CXIII.

Tabla CXIII. Valores estimados para las métricas del proyecto de prueba.

Métrica	Método	Entidad IS	SubElemento IS	Estrategia	C 1	C2	C3
Duración	Experiencia personal	Proyecto	-----	Abajo-arriba	6meses	6meses	6meses
Requerimientos terminados por ciclo	Experiencia personal	Producto	Requerimientos	Arriba-abajo	.r3, r7, r8,	r1, r2, r4 r5	r3. r6, r9, r10 r11, r12
Estabilidad	Experiencia personal	Producto	Requerimientos	Abajo - Arriba	3	5	0
Diseño validado	Experiencia Personal	Producto	Diseño	-----	3	4	6
Implementación por requerimiento	Experiencia personal	Producto	Pruebas	Arriba-abajo	3	4	6
Pruebas y requerimientos asociados	Experiencia personal	Producto	Pruebas	Abajo-arriba	3	4	6

Los valores actuales para los valores estimados se muestran en la Tabla CXIV.

Tabla CXIV. Valores actuales para las estimaciones de las métricas del proyecto de prueba.

Métrica	Entidad IS	SubElemento IS	Ciclo 1	Ciclo 2	Ciclo 3
Duración	Proyecto	-----	6meses	6meses	6meses
Requerimientos terminados por ciclo	Producto	Requerimientos	.r3, r7,	r8, r1, r2	r4. r5,
Estabilidad	Producto	Requerimientos	8	6	4
Diseño validado	Producto	Diseño	2	3	2
Implementación por requerimiento	Producto	Pruebas	2	3	2
Pruebas y requerimientos definidos	Producto	Pruebas	1	3	0

B.6 Anomalías

Las anomalías que se presentan son las del primer ciclo de trabajo y estas se muestran en la Tabla CXV.

Tabla CXV. Anomalías que se presentan para el primer ciclo de trabajo.

Métrica	Planeado	Obtenido	Ciclo
Duración	Proyecto	-----	Ciclo 1
Requerimientos terminados por ciclo	4	3	Ciclo 1
Estabilidad	3	8	Ciclo 1
Diseño validado	2	3	Ciclo 1
Implementación por requerimiento	2	3	Ciclo 1
Pruebas y requerimientos definidos	1	3	Ciclo 1

B.7 Interpretación

Las interpretaciones de las anomalías se muestran en la Tabla CXVI.

Tabla CXVI. Causas de las anomalías

Interpretación	
Anomalía	Interpretación
Requerimientos terminados por ciclo < planeado	Requerimientos mal definidos.
Estabilidad > planeado	

B.8 Retroalimentación

Las acciones de retroalimentación para las interpretaciones se describen en la Tabla CXVII.

Tabla CXVII. Acciones de retroalimentación para las interpretaciones halladas.

Retroalimentación	
Interpretación	Acción de retroalimentación
Requerimientos mal definidos.	Cambiar ingeniero análisis (corto plazo) Cambiar proceso captura requerimientos (largo plazo)

Apéndice C. Manual de Instalación del sistema SARE

En este apéndice se describe el procedimiento para la instalación de las herramientas utilizadas por SARE.

Pasos a seguir:

1. Instalar Java JDK 1.2+.
2. Instalar Apache Tomcat 4.1+.
3. Instalar MySQL.
4. Instalar CeWolf
5. Descargar e instalar el paquete SARE.zip

1. Instalar Java JDK

Pasos de instalación:

1. Descargar el JDK/SDK para la plataforma Windows en el sitio:
2. <http://java.sun.com/j2se/1.2.2/download.html>
3. Seleccione el icono del archivo descargado (j2sdk1.4_03.exe) para iniciar la instalación.
4. Siga las indicaciones de instalación.

2. Instalar Apache Tomcat.

Antes de iniciar la instalación de Apache Tomcat 4.1+ es necesaria la instalación del JDK.

Pasos de Instalación:

1. Descargar Tomcat del sitio: <http://jakarta.apache.org/tomcat/>
2. Seleccione el icono del archivo para descargar jakarta-tomcat 1.4.exe
3. Siga las indicaciones de instalación para definir el directorio de instalación.
4. Elija una contraseña para el administrador.

3. Instalar MySQL

Aunque MySQL no requiere JDK directamente asegúrese que esté se encuentre instalado, esto para realizar conexiones hacia MySQL.

Pasos de instalación:

1. Descargar la versión 4.0.x de MySQL en: <http://www.mysql.com/downloads>
2. También es necesario descargar el manejador de Java empleado para conectarse de un ambiente Java hacia MySQL, este se puede encontrar en: <http://www.mysql.com/downloads/api-jdbc-stable.html>
3. Descomprima el archivo MySQL.
4. Seleccione la opción "Setup.exe" para iniciar el proceso de instalación.
5. Siga las instrucciones de instalación y seleccione el directorio donde desea instalar MySQL.
6. Arranque la Base de Datos MySQL a través del Administrador winmysqladmin, esta herramienta reside en el directorio bin de la instalación MySQL.
7. Instalar el manejador en el directorio common\lib de la instalación del tomcat.

8. Instalar los scripts de configuración de tablas que vienen con el directorio de instalación de SARE, estos se encuentran en el directorio bin\scripts del directorio de trabajo de mysql.

4. Instalación de CeWolf

1. Ir a la sección dowloands y descargar cewolf1-0.
<http://cewolf.sourceforge.net/new/index.html>
2. Siga las instrucciones de instalación

5. Instalar SARE en el servidor

Pasos de Instalación

1. Descomprimir el archivo sare.zip
2. Añadir el paquete sare dentro de la siguiente ruta: C:\tomcat\medicion\<sare aquí>