



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

“MÉTODO Y HERRAMIENTA DE APOYO PARA MEDICIÓN Y
ANÁLISIS DE DATOS PARA FÁBRICAS DE SOFTWARE”

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERÍA
(COMPUTACIÓN)

P R E S E N T A:

MIGUEL AGUSTÍN RUIZ VELASCO SOBRINO

DIRECTOR DE TESIS:

M. en C. MARÍA GUADALUPE ELENA IBARGÜENGOITIA GONZÁLEZ

M. en C. CECILIA PEREZ COLÍN

MÉXICO, D.F.

2009.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicado

a mis Familia y Amigos

a Dios

Agradecimientos

Quiero agradecer a mi prometida Katya por haberme apoyado (y aguantado) antes y durante el posgrado, por haber permanecido junto a mi en las temporadas buenas y en las temporadas malas, sin cuyos ánimos tal vez no me habría decidido a dejar mi rutina anterior aventurándome en lo desconocido y me hubiera costado un trabajo casi infinito terminar; por tener alguien que me escuchara y me entendiera (y entendiera de lo que hablaba), aunque la conversación (y las quejas) fueran las mismas semana tras semana.

También quiero agradecer a mis padres Emma y Miguel, al Profesor Antonio, a mis hermanas Lola y Emma, y a los 6 gatos de la casa, por los ánimos que me dieron, por estar ahí a pesar de mi mal humor y recordarme que me quieren, que valía la pena el esfuerzo aunque yo estuviera matando al mundo. Incluso a Iván y su eternamente virulenta computadora, que mientras la limpiaba me levantaba el ánimo con sus elogios.

Otra que merece las más sinceras gracias es mi gran familia, preguntándome siempre como iba la tesis expresando tácitamente su apoyo.

A los compañeros del posgrado, víctimas del mismo mal, que han estado ahí para intercambiar una conversación a veces casual, a veces trascendente, para espantar un rato la soledad, que me siguieron invitando a ir con ellos a ir por un café a diseño a pesar que las repetidas negativas. Especialmente con los que más amistad hice: Andrés, Mari, Michel, Victor, Rubén, Sebastián, Magda, Tzolkin, Paty, espero no olvidar a nadie. También a Ely que siempre me decía: “ya apurate, o antes que te des cuenta vas a estar como yo de presionado con la tesis atrasada”, su consejo no cayó del todo en saco roto.

A mis amigos, unos más lejanos, otros más cercanos, siempre ofrecieron su amistad incondicionalmente, una palabra de aliento y un “¿qué? ¿todavía no terminas la tesis?, ¿pues qué rayos estás haciendo? ¡ya apúrate!”, como Elias que no ha podido entender porque me tardé tanto. A Lalo, que andaba igual que yo queriendo hacer una maestría, y que fue, junto con otros, un factor decisivo para dar el paso. Desde otra trinchera Toño y David que me animaron. En su momento me decían que era una oportunidad única en la vida, sin que yo valorara la sabiduría que sólo puede adquirirse con los años. Ahora veo la razón que tenían en su justa dimensión. También a Luis Roberto que durante la temporada con más presión me escuchaba en mis quejumbres, mientras yo oía sus planes de boda. Y a Karla y Roberto, por haber tenido y compartido la inquietud de aventurarse a una empresa que tiene un futuro prometedor.

Quiero agradecer a mis maestros, antiguos y recientes, que formaron con ahinco y dedicación las bases gracias a las cuales ésto hoy es posible y cuyos esfuerzos no siempre son valorados y raramente agradecidos. También a los sinodales que leen, revisan y aportan a las tesis que les asignan, por puro amor al arte y al avance de la ciencia. Y especialmente a mis tutoras, Lupita y Cecilia, que les tocó ver más de cerca el proceso de la elaboración de esta tesis, el cual fue algo tortuoso y cambiante, pero el esfuerzo ha valido la pena. Igualmente agradezco a las asistentes del Posgrado: Lulú, Diana, Amalia y Guadalupe, por chiquearnos y hacer por nosotros los trámites que los alumnos de otras instituciones tienen que hacer personalmente.

Y como al nombrar estoy omitiendo, quiero agradecer a muchos los que no he nombrado aquí, que seguramente dieron alguna palabra de aliento en un instante de soledad, aportaron de forma desinteresada una idea que después quedó plasmada, y que su intervención ha quedado olvidada. A todos el más profundo agradecimiento.

Finalmente quiero agradecer a Dios por haberme dado la vida y las abundantes oportunidades que he podido y querido (o no) aprovechar hasta llegar a este momento.

Índice general

Agradecimientos	III
Introducción	XV
1. Conceptos básicos de medición	1
1.1. Las mediciones en la vida ordinaria	1
1.2. ¿Que es la medición?	1
1.3. Las bases para la medición	2
1.4. La teoría de la representación de la medición	2
1.4.1. Relaciones empíricas	2
1.4.2. Las reglas de los mapeos	2
1.4.3. Pasos para un mapeo	3
1.5. Modelos y mediciones	3
1.5.1. Definiendo atributos	3
1.5.2. Mediciones directas e indirectas	3
1.5.3. Medición para predecir	3
2. La Medición en Ingeniería de Software	5
2.1. Aspectos de la medición en ingeniería de software	5
2.1.1. Negación a medir en la ingeniería de software	5
2.1.2. Hacer las cosas medibles	6
2.1.3. Objetivos de las mediciones de software	6
2.1.4. Las mediciones para entender, controlar y mejorar	7
2.2. Medición: Conceptos y prácticas básicas	7
2.2.1. Motivación para realizar mediciones	8
2.2.2. Las mediciones como un discriminador organizacional	8
2.2.3. Las mediciones de proyecto como base	9
2.2.4. Modelo de información de mediciones	10
2.2.5. Modelo de proceso de medición	10
2.3. El alcance de las mediciones de software	11
2.3.1. Estimación de costo y esfuerzo	11
2.3.2. Modelos y medidas de productividad	12
2.3.3. Recolección de datos	12
2.3.4. Modelos y medidas de calidad	12
2.3.5. Modelos de confiabilidad	12
2.3.6. Modelos y evaluaciones de desempeño	12
2.3.7. Mediciones de complejidad y estructura	12
2.3.8. Evaluación de madurez de capacidades	12
2.3.9. Modelos y evaluación del rendimiento	13
2.3.10. Administración por métricas (management by metrics, quantitative management)	13
2.3.11. Evaluación de Modelos y herramientas	13

2.4.	Definición de las entidades a medir	13
2.4.1.	Clasificación de las mediciones de software	13
2.4.1.1.	Procesos	14
2.4.1.2.	Productos	14
2.4.1.3.	Recursos	15
2.4.2.	GQM como técnica para definir mediciones	16
2.4.3.	Medición y mejora de procesos	17
3.	Proceso de Medición	19
3.1.	Plan de medición	19
3.1.1.	Identificar las necesidades de información	19
3.1.2.	Priorizar las necesidades de información	20
3.1.3.	Describir el contexto del proyecto	21
3.1.4.	Seleccionar las mediciones aplicables	21
3.1.5.	Especificar los indicadores	22
3.1.6.	Integrar el enfoque de medición en los procesos del proyecto	22
3.1.7.	Desarrollar procedimientos de medición	23
3.1.8.	Documentar el plan de medición del proyecto	24
3.2.	Realizar la medición	24
3.2.1.	Recolectar y procesar los datos	24
3.2.2.	Analizar los datos	25
3.2.3.	Generación de indicadores	25
3.2.4.	Representación de los indicadores	25
3.3.	Técnicas de análisis	26
3.3.1.	Estimación	26
3.3.1.1.	Proceso de estimación	26
3.3.2.	Análisis de factibilidad	26
3.3.2.1.	Indicadores de factibilidad	26
3.3.2.2.	Proceso de análisis de factibilidad	27
3.3.3.	Análisis de desempeño (performance)	27
3.3.3.1.	Indicadores de desempeño	27
3.3.3.2.	Proceso de análisis de desempeño	28
3.3.4.	Hacer recomendaciones	28
4.	El apoyo a la toma de decisiones	31
4.1.	Los objetivos de DSS	31
4.2.	Características de los problemas poco estructurados	31
4.3.	Estrategia de evolución de un DSS	32
4.4.	Mejoras esperadas del uso de un DSS	32
4.4.1.	Mejorar la efectividad de la toma de decisiones	32
4.4.2.	Mejora de la eficiencia de la toma de decisiones	32
4.4.3.	Mejorar la comunicación y colaboración entre las personas que toman las decisiones	33
4.4.4.	Mejorar el proceso de aprendizaje de los usuarios	33
4.5.	Posibles problemas derivados del uso de un DSS	33
4.5.1.	Restricción de las elecciones	33
4.5.2.	Asumir relevancia	33
4.5.3.	Transferencia de poder no intencional	34
4.5.4.	Obscurecimiento de la responsabilidad	34

5. La administración y el reuso de la experiencia	35
5.1. Beneficios del reuso de experiencia	35
5.2. Procesos de la administración de la experiencia	35
6. Relaciones de la medición con el funcionamiento de los equipos de desarrollo	37
6.1. Aplicación de la medición en el trabajo de los equipos	37
6.1.1. Tipos de organización	37
6.1.2. Estilos gerenciales	38
6.1.3. La teoría Z	39
6.1.4. La participación en la empresa	39
6.1.4.1. Participación en la información	40
6.1.4.2. Participación en la consulta	40
6.1.4.3. Participación funcional	41
6.1.4.4. El contexto del empowerment	41
6.1.4.5. Facultamiento (empowerment)	42
6.1.4.6. Participación en las utilidades	42
6.1.4.7. Participación en la propiedad	43
6.1.4.8. Participación institucional	43
6.1.5.	43
6.2. Establecer y sostener el compromiso	43
6.2.1. Tener el compromiso de la organización	43
6.2.2. Definir las responsabilidades en la medición	44
6.2.3. Proveer recursos	45
6.2.4. Capacitación en medición	45
6.2.5. Herramientas de medición	46
6.3. Ventajas de la adopción de la medición	46
6.4. Criterios de éxito de un programa de medición	47
7. Objetivos de la herramienta	49
7.1. Antecedentes	49
7.1.1. Aspectos técnicos	49
7.1.2. Aspectos humanos	50
7.1.3. Aspectos organizacionales	50
7.2. La transformación que ha traído Internet	51
7.2.1. Todos pueden publicar, y todos lo harán	51
7.2.2. Cuando hay datos abundantes, se privilegia la inteligencia	52
7.3. Objetivos de la herramienta	52
7.3.1. Propiciar la curiosidad y la búsqueda de alternativas	52
7.3.2. Conservar y compartir conocimiento	53
7.3.3. Motivación de los miembros de los equipos al participar de las decisiones	53
7.3.4. Balancear el control jerárquico vs. la independencia de los equipos	53
7.3.5. Integración con otras herramientas de recolección	53
7.3.6. Integración con otras herramientas de análisis	54
7.3.7. Integración con otros repositorios de conocimiento	54
7.3.8. Fácil acceso independientemente del dispositivo	54
7.4. Desarrollo de la herramienta	54
7.4.1. Descripción de la herramienta	54
7.4.2. Caso de uso general del sistema	55
7.4.3. Ejemplo de detalle algunos casos de uso	55
7.4.3.1. CRUD Proyectos	56

7.4.3.2.	Seleccionar proyecto	57
7.4.3.3.	Añadir proyecto	57
7.4.3.4.	Convertir datos	58
7.4.3.5.	Exportar datos	58
7.4.3.6.	Importar datos	59
7.5.	Tecnologías usadas	59
7.5.1.	Inyección de dependencias	60
7.5.1.1.	Componentes y servicios	60
7.5.2.	Spring Framework	61
7.5.3.	Hibernate	62
7.5.4.	Patrón de arquitectura MVC	63
7.5.5.	Apache Struts2	63
7.5.6.	AJAX	64
7.5.7.	Programación orientada a aspectos y AspectJ	65
7.5.8.	Sistemas basados en el conocimiento	66
7.5.8.1.	Arquitectura	67
7.5.9.	Drools	68
7.6.	Arquitectura	68
7.7.	Diagrama de clases	69
7.7.1.	Modelo de datos	69
7.7.2.	Acceso a datos	70
7.7.3.	Capa de Interfaz de usuario	72
7.7.3.1.	La recolección de eventos de calendario y los aspectos	74
7.7.4.	Base de datos	77
8.	Manual de usuario	81
8.1.	Pantalla de Bienvenida	81
8.2.	Entrada al sistema	81
8.3.	Seleccionar proyecto	82
8.4.	Metas	83
8.5.	Preguntas	85
8.6.	Métricas	86
8.7.	Proyectos	87
8.8.	Cuestionarios	90
8.9.	Recordatorios	92
8.10.	Tipos	96
8.11.	Procesos	97
8.12.	Respuestas	98
8.13.	Análisis de datos gráfico	100
8.13.1.	Entrada	100
8.13.2.	Generar un nuevo análisis OLAP	102
8.13.3.	Reorganizar dimensiones	102
8.13.4.	Generando una gráfica	103
8.13.5.	Guardar el análisis	105
	Conclusiones	107
	Bibliografía	109
A.	Casos de Uso	111
A.1.	Caso de uso general	111

A.2. Caso de uso CRUD GQM	111
A.2.1. CRUD Meta	112
A.2.1.1. Nueva Meta	113
A.2.1.2. Editar Meta	114
A.2.1.3. Borrar Meta	114
A.2.2. CRUD Pregunta	115
A.2.2.1. Nueva Pregunta	116
A.2.2.2. Editar Pregunta	116
A.2.2.3. Borrar Pregunta	117
A.2.3. CRUD Métrica	117
A.2.3.1. Nueva Métrica	118
A.2.3.2. Editar Métrica	119
A.2.3.3. Borrar Métrica	119
A.3. Caso de uso Parámetros	120
A.3.1. CRUD Procesos	121
A.3.1.1. Nuevo Proceso	122
A.3.1.2. Editar Proceso	122
A.3.1.3. Borrar Proceso	123
A.3.2. CRUD Tipos	124
A.3.2.1. Nuevo Tipo	125
A.3.2.2. Editar Tipo	125
A.3.2.3. Borrar Tipo	126
A.3.3. Calendario	127
A.3.3.1. Nuevo Evento del Calendario	128
A.3.3.2. Editar Evento del Calendario	128
A.3.3.3. Borrar Evento del Calendario	129
A.4. CRUD Proyectos	130
A.4.0.4. Seleccionar proyecto	131
A.4.0.5. Añadir proyecto	131
A.4.0.6. Editar proyecto	132
A.4.0.7. Borrar proyecto	132
A.4.0.8. Convertir datos	133
A.4.0.9. Exportar datos	133
A.4.0.10. Importar datos	134
A.5. Caso de uso Entrada de datos	134
A.5.1. Llenar registro	134
A.5.2. Ver registros	135
B. La hoja misteriosa	137
C. Compatibilidad con dispositivos móviles	139
C.1. Sistema	140
C.2. Reportes	140

Índice de figuras

2.1. Modelo de proceso de medición	11
7.1. Diagrama general de Casos de uso	55
7.2. Caso de uso de proyectos	56
7.3. Diagrama de componentes del sistema	68
7.4. Diagrama de clases del modelo de datos	70
7.5. Diagrama de clases de Acceso a datos	71
7.6. Diagrama de clases de ETL	72
7.7. Diagrama de clases de servicios	72
7.8. Capa de interfaz: GoalAction	73
7.9. Capa de interfaz: ProjectAction	73
7.10. Diagrama de base de datos	78
7.11. Diagrama de base de datos del planificador	79
7.12. Diagrama de base de datos del data warehouse	79
7.13. Diagrama de base de datos de autenticación	80
8.1. Pantalla de Bienvenida	81
8.2. Entrada al sistema	82
8.3. Seleccionar proyecto	82
8.4. Proyecto seleccionado	83
8.5. Pantalla principal de metas	83
8.6. Detalle de una meta	84
8.7. Insertar meta	84
8.9. Pantalla principal de preguntas	85
8.8. Editar meta	85
8.10. Editar pregunta	86
8.11. Pantalla principal de métricas	86
8.12. Editar métrica	87
8.13. Pantalla principal de proyectos	88
8.14. Insertar proyecto	88
8.15. Editar proyecto	89
8.16. Exportar datos a un DataWareHouse	89
8.17. Exportar proyecto	90
8.18. Pantalla principal de cuestionarios	90
8.19. Insertar cuestionario	91
8.20. Editar cuestionario	92
8.21. Programar una serie de recordatorios sobre un cuestionario	93
8.22. Programar una serie de recordatorios sobre un cuestionario	94
8.23. Programar una serie de recordatorios sobre un cuestionario	95
8.24. Listado de reportes pendientes de llenar	95
8.25. Responder un reporte pendiente directamente	96
8.26. Pantalla principal de tipos	96
8.27. Editar tipo	97

8.28. Pantalla principal de procesos	97
8.29. Editar proceso	98
8.30. Elegir cuestionario para responder	98
8.31. Responder cuestionario	99
8.32. Portada del data-warehouse	100
8.33. Entrada al data-warehouse	101
8.34. Escritorio del data-warehouse	101
8.35. Interfaz para definir un nuevo análisis usando cubos OLAP	102
8.36. Primer resultado del análisis	102
8.37. Organización original de las dimensiones del cubo	103
8.39. Resultado de reorganizar las dimensiones del cubo	103
8.38. Reorganizando las dimensiones del cubo	104
8.40. Opciones de gráfica	104
8.41. Comparación entre tiempo capturado y tiempo del reporte por proceso	105
8.42. Guardar el cubo para volverlo a usar	105
A.1. Diagrama general de Casos de uso	111
A.2. Caso de uso GQM	112
A.3. Caso de uso: CRUD meta	112
A.4. Caso de uso: CRUD pregunta	115
A.5. Caso de uso CRUD métricas	117
A.6. Caso de uso de parámetros del sistema	120
A.7. Caso de uso CRUD procesos	121
A.8. Caso de uso de tipos	124
A.9. Caso de uso de calendario	127
A.10. Caso de uso de proyectos	130
A.11. Caso de uso entrada de datos	135
C.1. Página de Bienvenida	141
C.2. Página principal y menú	141
C.3. Página de Proyectos	141
C.4. Página de métricas	141
C.5. Detalle de proyecto	142
C.6. Llenado de cuestionario	143
C.7. Entrada a los reportes	144
C.8. Página de reportes	144
C.9. Reporte	144
C.10. Reporte con zoom	144

Índice de cuadros

2.1. Ejemplos de componentes de la medición de software	14
---	----

Introducción

Casi cualquier sistema construido en la actualidad contiene cierta cantidad de software, que está creciendo en tamaño, complejidad y costo para desarrollar y cumple un rol importante en la productividad, eficiencia y seguridad de los sistemas. Esta tendencia ha provocado una mayor necesidad de formas de medir y administrar la calidad de el proceso con el que se desarrolla el software y sus productos resultantes.

Las métricas de software son algunas de las herramientas de las que los desarrolladores de software disponen para proveer de medidas efectivas de medición y administración de proyectos de software. Usar métricas de software para medición permite a una organización evaluar el software producido para asegurarse continuamente durante el proyecto que se adhiere a los lineamientos de productividad originales.

También el uso de métricas de software, ayuda a los administradores del proceso de software a evaluar seguimiento del desarrollo contra planes y calendarios. Finalmente la calidad inherente de un producto de software y la eficiencia de un desarrollo se pueden identificar, cuantificar y mejorar.

La implementación de un programa inicial de métricas tiene cuatro componentes:

- Definir un conjunto inicial de medidas a tomar
- Convencer que se tomen estas medidas
- Usar herramientas para reunir automáticamente (o lo más automáticamente posible) las medidas en métricas
- Publicitar los éxitos y apoyar el libre intercambio de ideas

Es necesario que las organizaciones empiecen poco a poco con un programa de métricas en vez de comenzar de lleno en un programa complejo desde el principio.

Los criterios que influyen seriamente el éxito de un programa de métricas:

- Seleccionar medidas que respondan a las necesidades de la organización
- Elegir medidas que no obstruyan las necesidades ni los procesos de la organización
- Saber que esperar de esas medidas
- Convencer a la alta dirección que la medición es necesaria y útil

Un programa de métricas de software completamente integrado en un entorno de Ingeniería de software como los de una empresa de nivel 4 o 5 CMMI es extremadamente complicado. Esto podría indicar porque demasiadas organizaciones tratan de alcanzar más de lo que es posible para su nivel actual de madurez, lo que hace fracasar el esfuerzo. El énfasis debería de estar puesto en empezar con poco y trabajar hacia conseguir metas organizacionales más avanzadas. Muchas organizaciones fracasan al implementar un programa exitoso de métricas, debido a que comienzan con un programa demasiado ambicioso que exige demasiada madurez y dedicación del personal y de los procesos.

Las métricas son especialmente útiles para una organización con centros de trabajo distribuidos, ya sea a través del mundo, o simplemente trabajando en las oficinas de varios clientes. La falta de contacto cotidiano entre los miembros de diferentes equipos de desarrollo dificulta la coordinación

Introducción

de la organización en su conjunto. Las métricas pueden jugar un papel importante en el esfuerzo de suplir esta carencia.

Por otro lado, resulta más fácil mantener operando un programa de métricas si se involucra a los participantes más allá de ser meros recolectores de información para convertirlos también en usuarios de ésta. También esto podría ayudar a madurar al equipo en su capacidad de auto gestión.

En una primera fase, las métricas seleccionadas deberán de ser fáciles de tomar; el éxito de un programa de métricas depende fuertemente en la exhaustividad y exactitud de los datos reunidos, que a su vez depende fuertemente del compromiso de la gente que reúne los datos.

La gente debe de estar convencida de la importancia de reunir los datos y no la deberá de asustar el proceso riguroso para recabarlos. El uso de herramientas automáticas puede aminorar el temor e incrementar la exactitud y consistencia de los datos colectados.

De hecho, en la mayoría de las organizaciones se recaban algunas métricas básicas, pero la gente no se percata que son métricas ni del uso que les pueden dar.

La presente tesis está organizada de la siguiente forma:

Una exposición breve de conceptos básicos de medición en general.

Un acercamiento a las mediciones desde un enfoque de Ingeniería de Software, sus alcances y utilidad.

Una breve introducción a los sistemas de soporte a decisiones y a los sistemas de soporte a decisiones basados en el conocimiento.

Una explicación de la participación de los colaboradores de la organización y sus efectos, así como otros temas de factor humano.

Una breve introducción a la administración de la experiencia así como algunas de sus posibilidades.

Un planteamiento del problema, así como de las condiciones que hacen más propicia la solución propuesta en esta tesis.

Una descripción de la herramienta propuesta.

Un manual de usuario de la herramienta propuesta.

Finalmente, conclusiones y trabajos futuros.

1. Conceptos básicos de medición

Las métricas de software, alguna vez una disciplina obscura y esotérica, se ha convertido en una parte esencial de la buena ingeniería de software. Muchos de los mejores desarrolladores de software miden características del software para adquirir una noción de si el diseño es de calidad y si el código está listo para comenzar las pruebas. Los buenos administradores de proyectos de software miden atributos del proceso y los productos para saber si el software estará listo para ser entregado y si el presupuesto será excedido. Por su parte, los clientes analíticos miden aspectos del producto final para determinar si cumple los requerimientos y es de suficiente calidad. Y los mantenedores deben ser capaces de evaluar el producto para saber qué tendrá que ser mejorado o actualizado.

1.1. Las mediciones en la vida ordinaria

La medición es el corazón de muchos sistemas que hacen posible nuestras vidas. Las mediciones económicas determinan los precios y los incrementos de los salarios; en sistemas de radar las mediciones permiten detectar aviones cuando es imposible verlos directamente; en los sistemas médicos permiten a los médicos diagnosticar enfermedades específicas. Sin mediciones la tecnología no puede funcionar.

Pero las mediciones no son sólo del dominio de los profesionales de la tecnología. Cada uno las usa en su vida diaria. Los precios actúan como una medida del valor de las cosas en una tienda y calculamos el total de la cuenta para asegurarnos que nos dan la cantidad correcta de cambio. Usamos medidas de peso y estatura para asegurarnos que la ropa nos quedará correctamente. Al hacer un viaje, calculamos la distancia, elegimos una ruta, medimos la velocidad y predecimos cuando llegaremos a nuestro destino (y tal vez si necesitaremos cargar gasolina). De esta forma, las mediciones nos permiten comprender el mundo, interactuar con nuestro entorno y mejorar nuestra vida.

1.2. ¿Que es la medición?

Estos ejemplos presentan una imagen de la variedad de los usos de las mediciones. Pero hay un común denominador en todas las actividades anteriormente descritas: en cada caso se asigna un descriptor a un aspecto del asunto que nos permite compararlo con otros.

Medición: es el proceso mediante el cual se asignan números o signos a atributos de entidades del mundo real de forma tal que los describen de acuerdo a reglas claramente definidas. [FP97]

Por lo tanto, las mediciones capturan información acerca de atributos de entidades. Una entidad es un objeto (como una persona o una habitación) o un evento (como un viaje o las pruebas de una fase de un proyecto de software) en el mundo real. Queremos describir la entidad identificando características que nos son importantes para distinguir una entidad de otra. Un atributo es una característica o propiedad de una entidad. Entre los atributos más comunes están el color o área (de un cuarto), el costo (de un viaje), o el tiempo usado (por la fase de pruebas). Es incorrecto decir que medimos cosas o que medimos atributos; de hecho, medimos atributos de cosas. Es ambiguo decir que “medimos un cuarto”, porque podemos medir su longitud, área o temperatura. También es ambiguo decir que “medimos la temperatura”, porque medimos la temperatura de un lugar geográfico específico bajo condiciones específicas. En otras palabras, lo que es común en el lenguaje común, es impropio para los científicos o ingenieros. Cuando describimos entidades usando atributos, normalmente describimos los atributos usando números o símbolos y al definir los símbolos o números, tratamos de preservar ciertas relaciones que observamos entre las entidades.

1.3. Las bases para la medición

Ordinariamente cuando medimos cosas, no pensamos en los principios científicos que estamos aplicando. Para hacer una medición usamos al mismo tiempo herramientas y principios que damos por hecho, pero estas técnicas e instrumentos sofisticados de medición se han desarrollado a lo largo de mucho tiempo, basados en un creciente entendimiento de los atributos que se miden.

Desafortunadamente, no existe todavía un entendimiento comparable de los atributos del software ni tenemos las herramientas sofisticadas asociadas. Preguntas que son relativamente fáciles de responder para entidades que no son software, son difíciles para el software.

1.4. La teoría de la representación de la medición

En cualquier actividad de medición, hay reglas que se deben seguir. Estas reglas nos ayudan a tener consistencia en nuestras mediciones y a tener bases para interpretar los datos. La teoría de la medición nos da las reglas, proveyendo los cimientos para desarrollar y razonar acerca de cualquier tipo de medición. Este enfoque basado en reglas es común en muchas ciencias.

1.4.1. Relaciones empíricas

La teoría de la representación de la medición busca formalizar nuestras intuiciones acerca de la forma en que el mundo funciona. Los datos que obtenemos como medidas deberían representar atributos de las entidades que observamos y las manipulaciones que les hacemos a los datos deberían de preservar las relaciones que observamos entre las entidades; de esta forma la intuición es el punto de partida de cualquier medición.

Podemos comenzar a entender el mundo usando relaciones relativamente poco sofisticadas que no requieren herramientas de medición. Una vez que desarrollamos un entendimiento inicial y hemos acumulado algunos datos, lo podremos medir de formas más sofisticadas y con herramientas especializadas. Analizar los resultados generalmente conduce a clarificar y reevaluar el atributo y a relaciones empíricas todavía más sofisticadas. De esta forma, se aumenta la exactitud y se mejora el entendimiento.

Formalmente definimos medición como un mapeo del dominio empírico al dominio formal. Por lo tanto, una medición es el número o símbolo asignado a una entidad por este mapeo para caracterizar un atributo. [FP97]

Algunas veces, las relaciones empíricas para un atributo no están acordadas, especialmente cuando reflejan preferencias personales.

En estos casos, los calificadores tienen nociones del atributo que desean medir, pero no siempre hay un entendimiento común.

1.4.2. Las reglas de los mapeos

Hemos visto como una medición se usa para caracterizar un atributo. Se empieza en el mundo real, estudiando una entidad y tratando de comprender más de ella; por lo tanto el mundo real es el dominio del mapeo, y el mundo matemático es el rango. Cuando mapeamos el atributo a un sistema matemático tenemos muchas opciones para el mapeo y para el rango. Podemos usar números reales, enteros o incluso un conjunto de símbolos no numéricos.

Queremos que el comportamiento de las mediciones en el sistema numérico sea el mismo que los elementos correspondientes en el mundo real, para que al estudiar los números, aprendamos algo acerca del mundo real; por lo tanto queremos que el mapeo conserve la relación. Esta regla se llama la **condición de la representación**. La condición de la representación afirma que un mapeo M debe mapear entidades a números y relaciones a relaciones numéricas de forma tal que las relaciones empíricas preserven y sean preservadas por las relaciones numéricas.

1.4.3. Pasos para un mapeo

- Identificar atributos de alguna entidad del mundo real.
- Identificar relaciones empíricas de los atributos
- Identificar relaciones numéricas correspondientes a cada relación empírica
- Definir mapeos de las entidades del mundo real a los números
- Verificar que las relaciones numéricas preserven y sean preservadas por los mapeos

1.5. Modelos y mediciones

En general un **modelo** es una abstracción de la realidad, que nos permite quitarle detalle y ver una entidad o un concepto desde una perspectiva particular. Existen muchos tipos de modelos: ecuaciones, mapeos o diagramas, por poner algunos ejemplos. Estos nos permiten mostrar como las partes del componente se relacionan entre sí, de forma que podamos examinar y entender estas relaciones y hacer juicios sobre ellas.

Se ha visto que la condición de representación requiere que cada medición esté asociada con un modelo de cómo la medición mapea las entidades y atributos del mundo real a los elementos del sistema numérico. Estos modelos son esenciales al tratar de comprender no sólo como se deriva la medición, sino como interpretar el comportamiento de los sistemas numéricos al regresar al mundo real. Pero también necesitamos modelos incluso antes de empezar con el proceso de medición. De hecho al efectuar una medición, estamos midiendo un modelo de la entidad, no a la entidad en sí, por lo tanto el modelo del mapeo debería estar suplementado con un modelo del dominio del mapeo, o sea, con un modelo de cómo la entidad se relaciona con sus atributos.

1.5.1. Definiendo atributos

Al medir, siempre existe el peligro de que nos enfoquemos demasiado en el sistema matemático formal, y descuidemos el empírico. Nos apresuramos a crear mapeos y a manipular números, sin poner suficiente atención en las relaciones entre las entidades y sus atributos en el mundo real.

1.5.2. Mediciones directas e indirectas

La medición directa de un atributo de una entidad no involucra a ninguna otra entidad o atributo. Al contrario, la densidad de un objeto físico se puede medir sólo indirectamente en términos de su masa y volumen; por lo tanto usamos un modelo para mostrar que la relación entre los tres es $densidad = \frac{masa}{volumen}$.

1.5.3. Medición para predecir

Al hablar de medir algo, usualmente significa que queremos evaluar alguna entidad que existe. Estas mediciones son muy útiles para comprender que existe ahora o qué ha pasado en el pasado. Pero en muchas circunstancias nos gustaría predecir un atributo de alguna entidad que todavía no existe.

Muchas veces necesitamos predecir cuánto costará un desarrollo, o cuánto tiempo y esfuerzo será necesario, para poder asignar los recursos apropiados al proyecto. Sería inaceptable simplemente esperar a que el proyecto termine y medir atributos de costo y calendario.

La distinción entre medir para evaluar y medir para predecir no siempre es clara. Por ejemplo, supóngase que usamos un globo para determinar la distancia entre Londres y Nueva York; esta medida indirecta nos ayuda a saber que tan lejos están las ciudades entre sí. Sin embargo, la misma

1. Conceptos básicos de medición

actividad se realiza cuando queremos predecir la distancia que viajaremos en un viaje futuro. Hay que notar que lo que efectuamos al evaluar distancias involucra al globo como un modelo del mundo real, y procedimientos de predicción que describen como usar el modelo.

En general, las mediciones para predicción siempre requieren un modelo matemático en que se relacione, los atributos a ser predichos con otros atributos que se puedan medir ahora. El modelo no necesita ser complejo para ser útil. Algunas veces el mismo modelo es usado tanto para evaluar como para predecir.

En el siguiente capítulo trataremos el tema de las mediciones desde un punto de vista de la Ingeniería de Software, habiendo hablado brevemente de las mediciones en general en este.

2. La Medición en Ingeniería de Software

2.1. Aspectos de la medición en ingeniería de software

La ingeniería de software es una serie de técnicas que aplican un enfoque de ingeniería a la construcción y soporte de productos de software. [FP97] Estas actividades incluyen la administración, costeo, planeación, modelado, análisis, especificación, diseño, implementación, pruebas y mantenimiento del software. Por “enfoque de ingeniería” nos referimos a que cada actividad está entendida y controlada, así que hay pocas sorpresas conforme el software es especificado, diseñado, construido y mantenido. Mientras que las ciencias de la computación aportan las bases teóricas para construir software, la ingeniería de software se enfoca en implementar el software de una forma controlada y científica.

La ingeniería de software ha hecho un trabajo admirable proveyendo funcionalidad segura, útil y confiable. Pero hay mucho espacio para mejorar. La literatura está llena con ejemplos de proyectos que han excedido sus presupuestos y calendarios. Peor aún, hay muchas historias de software que ha puesto vidas y negocios en peligro.

Los ingenieros de software hemos atacado estos problemas buscando continuamente nuevas técnicas y herramientas para mejorar los procesos y los productos. La capacitación ayuda a estos cambios para que los ingenieros de software estemos mejor preparados para aplicar las nuevas técnicas para desarrollar y dar mantenimiento; pero sólo con mejoras metodológicas no harán una disciplina de verdadera ingeniería.

2.1.1. Negación a medir en la ingeniería de software

Las disciplinas de ingeniería usan métodos que están basados en modelos y teorías, muchos de estos modelos se han generado usando el método científico. El método científico se apoya en la medición: medir las variables para diferenciar casos, medir los cambios de comportamiento, y medir las causas y efectos.

Pero las mediciones tradicionalmente se han considerado un lujo en la ingeniería de software:

- Fallamos al no asignarle metas medibles a nuestros productos de software
- **Principio de Gilb's de las metas difusas:** los proyectos sin metas claras, no alcanzarán claramente sus metas.
- Fallamos al comprender y cuantificar los componentes de costos de nuestros proyectos de software.
- No cuantificamos ni predecimos la calidad de los productos que producimos.
- Permitimos que nos convenzan con evidencia anecdótica de probar otra tecnología revolucionaria de desarrollo, sin hacer un estudio cuidadosamente controlado para determinar si la tecnología es efectiva y eficiente.

Cuando se hacen mediciones, normalmente se hacen infrecuente, inconsistente e incompletamente. Las mediciones incompletas pueden ser frustrantes para quienes quieren usar los resultados. Se dan unos resultados que sugieren un estudio, pero no se dice cómo se obtuvieron esos resultados, cómo se diseñaron y ejecutaron los experimentos, qué entidades se midieron ni cómo, ni cuáles son márgenes realistas de error. Sin esta información adicional permanecemos escépticos sin poder decidir si aplicar

los resultados a nuestra propia situación. Adicionalmente, no podemos hacer un estudio objetivo para repetir las mediciones en nuestro propio ambiente. De esta forma, la falta de mediciones en ingeniería de software se adiciona a la falta de una aproximación rigurosa.

Está claro por lo que se hace en otras disciplinas de ingeniería que las mediciones pueden ser efectivas, si no esenciales, haciendo más visibles las relaciones y características, evaluando la magnitud de los problemas y la búsqueda de solución a los problemas.

2.1.2. Hacer las cosas medibles

“Hay que hacer medible lo que es inmedible” Galileo Galilei (1564-1682)

Esta frase sugiere que uno de los objetivos de la ciencia es encontrar formas de medir atributos de las cosas que nos interesan. Implícitamente en esta afirmación está la idea de que las mediciones hacen los conceptos más visibles y por lo tanto más entendibles y controlables.

Para incrementar el rigor de las mediciones en la ingeniería de software, no debemos restringir el tipo o rango de mediciones que podemos hacer. Aunque algunos ingenieros de software sigan diciendo que algunos atributos importantes de software como confiabilidad, calidad, usabilidad y facilidad de mantenimiento son simplemente no cuantificables, habrá que tratar de medirlos para incrementar nuestro conocimiento.

Puede haber controversia cuando tratamos de capturar información cualitativa acerca de algún aspecto de la ingeniería de software. Diferentes expertos tendrán diferentes opiniones y muchas veces es imposible tener consenso.

2.1.3. Objetivos de las mediciones de software

Aún cuando un proyecto no está en problemas, las mediciones no son sólo útiles sino necesarias. De hecho no podemos saber si el proyecto está saludable si no hay alguna medición de su salud. Así que las mediciones son necesarias al menos para evaluar el estado de nuestros proyectos, productos, procesos y recursos. Pero como normalmente no sabemos que desequilibra un proyecto, es esencial que midamos y registremos características de proyectos buenos y de proyectos malos. Necesitamos documentar tendencias, la magnitud de acciones correctivas y los cambios resultantes. En otras palabras, debemos controlar nuestros proyectos, no sólo dejarlos andar.

Medir para decidir David Luján

No puedes controlar lo que no puedes medir DeMarco, 1982

Los objetivos de medición deben ser específicos, relacionados a lo que los administradores, desarrolladores y usuarios necesitan saber. Por eso, estos objetivos pueden ser diferentes dependiendo del rol de la persona involucrada y a que nivel del desarrollo de software se generan. Pero son las metas las que nos indicarán como se usará la información de las mediciones una vez que se obtenga.

Ejemplos de metas de algunos roles:

- Administradores
 - ¿Cuánto cuesta cada proceso?
 - ¿Qué tan productivo es el personal?
 - ¿Qué tan bueno es el código que se desarrolla?
 - ¿El usuario final quedará satisfecho con el producto?
 - ¿Como podemos mejorar?
- Ingenieros

- ¿Es posible probar los requerimientos?
- ¿Hemos encontrado todas las fallas?
- ¿Hemos cumplido nuestras metas de producto o proceso?
- ¿Que pasará en el futuro?

2.1.4. Las mediciones para entender, controlar y mejorar

Las listas anteriores muestran que la medición es importante para estas tres actividades básicas.

Primero, hay mediciones para entender que está sucediendo durante el desarrollo y mantenimiento. Podemos evaluar la situación actual, establecer líneas base (baselines) que nos ayuden a poner metas para un comportamiento futuro. De esta forma, las mediciones incrementan nuestra visibilidad hacia algunos aspectos de los procesos, dándonos más entendimiento de las relaciones entre las actividades y las entidades que afectan.

Segundo, las mediciones nos ayudan a controlar lo que está sucediendo en nuestros proyectos. Usando las líneas base, metas y el conocimiento de las relaciones, podemos predecir que es posible que suceda y hacer cambios a los procesos y productos que nos permitirán cumplir nuestras metas.

Tercero, las mediciones nos motivan a mejorar nuestros procesos y productos.

No importando como se usen las mediciones, es importante conocer y guiar las expectativas de aquellos que toman decisiones basadas en las medidas. Los usuarios de los datos siempre deberán estar concientes de la limitada precisión de las predicciones y del margen de error en las mediciones. Como en cualquier otra disciplina ingenieril, hay amplio espacio para abusar y mal usar los resultados en la ingeniería de software.

2.2. Medición: Conceptos y prácticas básicas

Todas las organizaciones exitosas de software implementan la medición como parte de sus actividades diarias técnicas o de management (gestión). Las mediciones permiten tener la información objetiva que se necesita para tomar decisiones informadas que impactan positivamente el desempeño de las actividades de negocio y de ingeniería. En las organizaciones de software exitosas la información derivada de las mediciones se trata como un recurso importante y se pone a disposición de las personas que toman las decisiones a todos los niveles.

Las mediciones de software han pasado a ser una disciplina esencial de la ingeniería de software. En el pasado, la medición se trataba como una tarea que no aportaba ningún valor adicional, una cosa más que hacer. Ahora se considera como una actividad básica de la ingeniería de software al ser incluida en el nivel 2 de CMMI.

- La forma en que las mediciones se implementan y se usan en una organización determinan cuanto valor se puede obtener en mejoras en sus procesos de ingeniería y de negocios. Las mediciones son más útiles cuando se implementan para soportar los objetivos técnicos y de negocio de una organización y cuando se integran con las actividades técnicas y de management que define un proyecto de software. La mayor utilidad de las mediciones es cuando se recolecta información objetiva relacionada con los problemas y riesgos que puede afectar los objetivos de un proyecto. Las mediciones funcionan mejor cuando se consideran una parte integral y significativa de la administración de proyectos.
- Las mejores organizaciones diseñan sus procesos técnicos y de management (gerenciales o administrativos) para usar los datos producto de mediciones objetivas. Los datos producto de las mediciones y los análisis que se derivan soportan la toma de decisiones a corto y a largo plazo. Una organización madura típicamente usa las mediciones para ayudar a planear y evaluar una propuesta de proyecto, para controlar el desempeño contra lo planeado, a guiar las decisiones

e inversiones de mejora de procesos, y para evaluar el desempeño técnico y del negocio contra los requerimientos de mercado. Una buena organización utiliza mediciones en todo el ciclo de vida del software, desde la concepción hasta el retiro; una medición se implementa como una disciplina proactiva, y la información derivada de las mediciones se considera un recurso estratégico.

2.2.1. Motivación para realizar mediciones

El software se ha convertido en una de las mayores inversiones y factores de estrategia de negocio, incluso para organizaciones no basadas en el software. Dadas las grandes inversiones en el desarrollo y mantenimiento de activos críticos de información, hay una demanda para una evaluación y administración más objetiva de proyectos con componentes importantes de software.

Las mediciones son más importantes en el nivel de proyectos. Las mediciones de software ayudan al administrador de proyecto a tomar mejores decisiones; ayuda a definir e implementar planes más realistas, a asignar más provechosamente recursos escasos, y a monitorear con exactitud el progreso y el desempeño de esos planes. Las mediciones dan la información requerida para tomar decisiones clave de proyecto y a tomar acciones apropiadas; a relacionar e integrar la información derivada de otros proyectos y otras áreas técnicas. En suma, permite al administrador de proyecto tomar decisiones usando información objetiva. Específicamente, las mediciones de software proveen información objetiva que permite:

- Comunicarse efectivamente, las mediciones permiten diseminar información objetiva a lo largo y ancho de la organización, e incluso fuera de ella.
- Dar seguimiento a objetivos específicos del proyecto, las mediciones pueden describir el status de los procesos y/o productos de un proyecto de software.
- Identificar y corregir problemas a tiempo, las mediciones facilitan una estrategia proactiva de management; fomenta el descubrimiento y la corrección de problemas técnicos o administrativos que puede ser más caro y difícil de corregir posteriormente, para evitar caer en una enfoque reactivo de corregir los problemas cuando pasan.
- Tomar decisiones de trade-off, cada proyecto tiene restricciones. Costo, calendario, capacidad, calidad técnica y rendimiento tienen que ser balanceados entre sí y administrados conjuntamente para cumplir con los objetivos de proyecto establecidos, las decisiones en un área afectan las otras, aunque no estén relacionadas.
- Justificar decisiones, los entornos de negocio actuales exigen un buen rendimiento en los proyectos; los gerentes técnicos, de proyecto y de negocio tienen que poder defender las bases de sus estimados y planes con datos históricos; justificar cambios a los planes con datos de productividad actualizados. Las mediciones permiten tener un argumento racional para elegir las mejores alternativas.

Como cualquier herramienta técnica o de management, las mediciones no pueden garantizar que un proyecto será exitoso, sin embargo ayudan al tomador de decisiones a tener una actitud proactiva para lidiar con los asuntos más importantes inherentes a los proyectos. Las mediciones ayudan a mejorar al proyecto, y por lo tanto, a la organización completa.

2.2.2. Las mediciones como un discriminador organizacional

El entorno actual de negocios se caracteriza por un mercado altamente competitivo y un cambio constante de la tecnología. Tanto en los mercados de software empaquetado como a la medida los clientes exigen más funcionalidad a menor precio y la implementación rápida de cualquier capacidad

para satisfacer sus necesidades de negocio. En el entorno actual es cada vez más difícil convertir una organización como un líder en el mercado de TI, y todavía más difícil mantenerla como una organización de alto rendimiento, o sea que funcione mejor que sus competidores tanto técnicamente cómo negocio.

La experiencia ha mostrado que casi cualquier organización de alto rendimiento se puede describir de la siguiente forma:

- Los tomadores de decisiones disponen de información objetiva, exacta y actualizada y es una parte integral de la cultura corporativa.
- Las perspectivas, tanto técnicas como de negocios, pasadas, presentes y futuras, son tomadas en cuenta para definir los objetivos y expectativas de los proyectos.
- Los procesos y procedimientos organizacionales, están diseñados para identificar, caracterizar y administrar el cambio; lidiar con él es parte de la forma de hacer negocios de la organización.
- Las noticias buenas o malas se comunican dentro de la organización. Los problemas se identifican y tratan abiertamente.
- Hay una preferencia para tomar decisiones y acciones basadas en información.

Todas estas características están basadas en la información, y por lo tanto en las mediciones. Para ser una organización de alto rendimiento en el sector de TI, la organización necesita de forma regular la información correcta, para tomar las decisiones correctas. Usa información para hacerse más eficiente y para producir productos de mejor calidad. Las mediciones facilitan y aceleran el aprendizaje organizacional y colaboran a la adaptación de la organización en el mercado; proveen una estructura para aprender de cada proyecto, haya sido buena o mala la experiencia; ayuda a comprender la diferencia entre su rendimiento y los niveles de rendimiento demandados por el mercado; permite que la organización mejore dentro de sus límites de negocio y técnicos. Con todo esto, la información derivada de la medición, se convierte en un recurso para competir y un proceso efectivo de medición se traduce en un discriminador entre organizaciones.

2.2.3. Las mediciones de proyecto como base

Dados los fuertes lazos entre la productividad de una organización de software y la medición, el reto es implementarla de la forma que tenga el mayor impacto positivo en cada uno de los proyectos dentro de la organización. La base de un programa de medición reside al nivel de proyectos, y el éxito de la organización es el éxito de sus proyectos. Para ser exitosa, los gerentes técnicos y de proyecto tienen que tomar continuamente importantes decisiones de los recursos del proyecto, el calendario y las características funcionales, tomando en cuenta restricciones organizacionales y de mercado, habiendo muy poco espacio para el retrabajo o para reiniciar.

El administrador de proyectos debe integrar todos éstos aspectos del software para ser exitoso. Se tienen que tomar decisiones diariamente o incluso cada pocas horas, de cómo se desarrollarán y administrarán los productos técnicos, como se asignarán los recursos, y qué asuntos se atacarán y con qué prioridad. En la medida que estas decisiones sean efectivas, también lo será el proyecto.

El entorno de los proyectos está formado de múltiples consideraciones técnicas, de costo y de calendario; la experiencia muestra que incluso en organizaciones pequeñas de tecnología, cada proyecto es único en términos de dominio de aplicación, procesos técnicos, interdependencias operativas con otros sistemas y otras restricciones. Sin mencionar el continuo cambio técnico y administrativo de forma que cada proyecto se perfila con diferentes características de proceso y de producto.

El enfoque organizacional de medición debe de ser adaptable y responder efectivamente a las necesidades de información de cada proyecto.

Aunque el nivel primario para implementar las métricas en una organización es a nivel de proyecto, también hay necesidades válidas de información en niveles más altos dentro de la organización. En casi todos los casos, la información a nivel de organización, se deriva de los proyectos. Se necesita buena información de proyectos para analizar la organización en su conjunto, que se puede ver como la medición de muchos proyectos al mismo tiempo, combinando de forma válida los datos generados por cada proyecto y usando diferentes técnicas de análisis para satisfacer diferentes necesidades de información. En todo caso, la principal variable es la disponibilidad de información de calidad a nivel de proyecto porque, de hecho, la mayoría de las actividades de medición en la empresa suceden a nivel de proyecto. [MCJ⁺02]

2.2.4. Modelo de información de mediciones

El modelo de información de medición es un mecanismo para relacionar las necesidades de información previamente definidas a los procesos y productos del proyecto que son las entidades que pueden ser medidas directamente y establece una base para comunicar efectivamente los resultados de la medición dentro de la organización.

También provee una estructura que define las medidas específicas del proyecto y las relaciona con las necesidades de información de los tomadores de decisiones.

Durante las fases de planeación y ejecución de un proyecto de software, se tienen que tomar continuamente decisiones técnicas y administrativas, así como trade-offs en costo, calendario, características y calidad; por esto existe una necesidad de información para soportar el proceso de toma de decisiones. El modelo de información de mediciones ayuda a definir las necesidades de información de los tomadores de decisiones y enfoca las actividades de planeación de medición en la selección y especificación de las medidas más apropiadas para satisfacer esas necesidades. Al irse implementando las medidas y empezando a reunir datos, el modelo estructura las mediciones y los análisis asociados en productos de información. Estos productos integran los resultados de las mediciones con los criterios establecidos de decisión y da recomendaciones de alternativas a los administradores de proyecto.

En un proyecto típico hay muchas necesidades de información en cualquier momento, estas necesidades tienden a cambiar significativamente durante el desarrollo del proyecto, basado en el cambio de objetivos, supuestos, y restricciones del proyecto.

2.2.5. Modelo de proceso de medición

El modelo está construido en una secuencia administrativa de “Planear-hacer-revisar-actuar”, adaptada para soportar las actividades y tareas específicas de las mediciones. El proceso de medición incluye cuatro actividades, que son esenciales para una implementación exitosa.

- Planear la medición
- Realizar la medición
- Evaluar la medición
- Establecer y sostener el compromiso

La planeación de la actividad comprende la identificación de las necesidades de información del proyecto y la selección de las mediciones para poder satisfacer las necesidades del modelo de información, también incluye las tareas de definición de los procesos de recolección de datos, análisis y reporte de los datos, la planeación de la evaluación de las mediciones dando como resultado varios productos de información, y la planeación de la evaluación del proceso en sí mismo. Como parte de este proceso se considera la implementación de las actividades de medición en los procesos técnicos y administrativos, así como los recursos y tecnologías requeridas para implementar un programa de mediciones

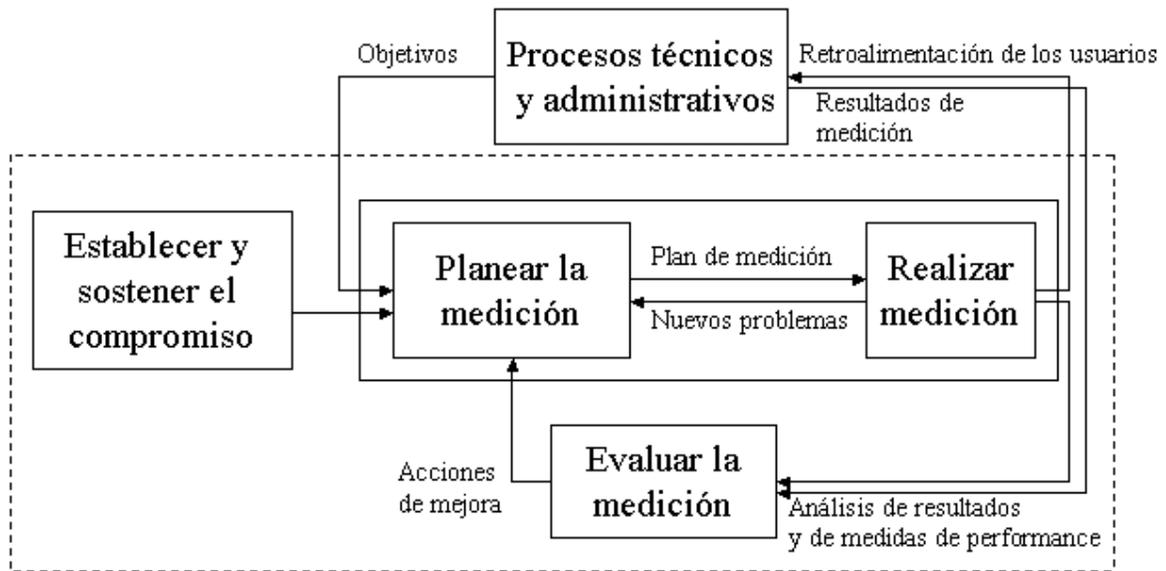


Figura 2.1.: Modelo de proceso de medición

para el proyecto. El resultado debe de ser un plan definido de medición acorde a las necesidades de información del proyecto.

La realización de las mediciones, junto con el plan, son las actividades centrales que satisfacen directamente los requerimientos de los usuarios de mediciones. Realizar mas mediciones conlleva la recolección y el procesamiento de los datos de mediciones, el uso de los datos para analizar las necesidades individuales de información y también como se interrelacionan y la generación de productos de información para presentar los resultados de los análisis, alternativas y recomendaciones a los tomadores de decisiones. Esta fase implementa el plan y produce la información necesaria para tomar decisiones basados en las mediciones.

En el proceso de evaluación de las mediciones, se analiza el proceso de medición en sí; el proceso evalúa tanto las mediciones aplicadas como la capacidad del proceso de medición y permite identificar áreas de mejora. Esta fase asegura que el proceso de medición está en constante actualización a las necesidades presentes de información y ayuda a elevar la madurez del proceso a nivel proyecto y organización.

La actividad de establecer y sostener el compromiso asegura que las actividades de medición son contempladas en los niveles de organización y de proyecto; y provee los recursos y la infraestructura organizacional requerida para implementar un programa viable.

El proceso de medición es iterativo por naturaleza, está definido para adaptarse a las características y al contexto de un determinado proyecto y para ser adaptable a la información y las decisiones cambiantes. Tanto el modelo de información como el proceso de medición deben de establecer un enfoque que captura la experiencia y las lecciones aprendidos de las iniciativas de medición anteriores, porque juntas deben ser el cimiento de un programa de medición exitoso.

2.3. El alcance de las mediciones de software

2.3.1. Estimación de costo y esfuerzo

Los administradores fueron los primeros motivadores en usar mediciones de software porque querían predecir los costos de un proyecto durante etapas tempranas del ciclo de vida del software.

2.3.2. Modelos y medidas de productividad

Las necesidades de los administradores también han resultado en numerosos intentos para definir medidas y modelos para evaluar la productividad durante diferentes procesos de software y en distintos entornos.

2.3.3. Recolección de datos

La calidad de cualquier programa de medición es claramente dependiente de una recolección cuidadosa de datos. Pero recolectar datos no es una tarea fácil, especialmente cuando la recolección debe ser hecha a través de una variedad de proyectos. Por esto la recolección de datos se está convirtiendo en una disciplina en sí misma, donde especialistas trabajan para asegurarse que las mediciones se definan sin ambigüedad, la recolección sea consistente y completa y la integridad de los datos no esté en riesgo.

2.3.4. Modelos y medidas de calidad

La productividad no puede ser analizada aisladamente. Sin una evaluación de calidad que la acompañe, no tiene sentido medir la velocidad de producción. Los modelos de calidad usualmente involucran mediciones de producto, dado que su más grande meta es predecir la calidad de los productos.

Los factores de calidad usualmente usados en la mayoría de los modelos de calidad, corresponden a atributos externos de producto. Los criterios con los cuales se clasifican los factores generalmente corresponden a atributos internos de productos o procesos. De esta forma las métricas que miden los criterios corresponden a medidas propuestas de los atributos internos; en cada caso, los términos usados son generales y usualmente no están bien definidos ni son precisos.

2.3.5. Modelos de confiabilidad

La mayoría de los modelos de calidad incluyen la confiabilidad como un factor, pero la necesidad de predecir y medir la confiabilidad ha dado lugar a una especialización propia.

2.3.6. Modelos y evaluaciones de desempeño

El desempeño es otro aspecto de la calidad. El trabajo que se ha hecho para la evaluación del desempeño va desde características externamente observables como tiempos de respuesta y tasas de terminación, hasta factores internos como la eficiencia de los algoritmos usando herramientas de complejidad computacional y algorítmica.

2.3.7. Mediciones de complejidad y estructura

Algunos atributos deseables como confiabilidad y facilidad de mantenimiento no pueden ser medidos hasta que haya una versión operativa del código. Aún así, queremos poder predecir que partes de un sistema de software es probable que sean menos confiables, más difíciles de probar o requieran más mantenimiento que otras, aún antes de que el sistema esté completo.

2.3.8. Evaluación de madurez de capacidades

Como se ha visto el modelo del SEI de madurez de capacidades, y sus derivados intentan capturar nociones de la visibilidad de procesos: claramente un atributo interno de proceso. Pero el nivel de madurez, también ha sido interpretado en una forma muy diferente, como un atributo de recursos.

La calificación de madurez se ve como un atributo de la empresa, no como del proceso de la empresa.

2.3.9. Modelos y evaluación del rendimiento

La evaluación del rendimiento está relacionada con medir la eficiencia, un atributo de producto. Este atributo puede implicar eficiencia en tiempo o en espacio. Como la confiabilidad la eficiencia es un atributo de producto prominente en las jerarquías de la mayoría de los modelos de calidad. Normalmente, una descripción a grandes rasgos del algoritmo es suficiente para dar predicciones suficientemente buenas de la eficiencia del código implementado.

2.3.10. Administración por métricas (management by metrics, quantitative management)

Los gerentes de proyecto frecuentemente usan métricas para fijar objetivos para sus equipos de desarrollo y muchas compañías y organizaciones definen un conjunto estándar de mediciones y métodos para reportar, de forma que se puedan comparar y contrastar los proyectos.

2.3.11. Evaluación de Modelos y herramientas

Comúnmente las organizaciones consideran invertir en una nuevo método o herramienta pero dudan por la incertidumbre acerca de los costos, utilidad o efectividad. Algunas veces la herramienta propuesta o el método se prueba primero en un proyecto pequeño y los resultados se evalúan para determinar si conviene hacer más inversiones y una implementación más amplia.

2.4. Definición de las entidades a medir

2.4.1. Clasificación de las mediciones de software

La primera tarea de cualquier actividad de medición de software es identificar las entidades y atributos que queremos medir. En software hay tres clases:

Procesos, son conjuntos de actividades relacionadas con el software.

Productos, son cualquier salida, entregable o documentación que resulta de una actividad de proceso.

Recursos, son entidades requeridas por una actividad.

Un proceso está usualmente asociado con algún intervalo de tiempo, o sea, cada una de las actividades de un proceso están ordenadas o relacionadas de alguna forma que involucra al tiempo, de forma que una actividad tiene que terminarse antes de que pueda comenzar otra. El ordenamiento puede ser implícito o explícito.

Los recursos y productos se asocian con los procesos. Cada actividad dentro de un proceso tiene recursos que emplea en su realización y también produce productos. Por esto la salida de una actividad podría ser entrada para otra.

En cada tipo de entidad, se pueden distinguir entre atributos externos e internos:

Los **atributos internos** de un producto, proceso o recurso, son los que pueden ser medidos puramente en términos del producto, proceso o recurso por sí mismo. En otras palabras, un atributo interno puede ser medido al examinar el producto, el proceso o el recurso en sí mismo, independientemente de su comportamiento.

Los **atributos externos** de un producto, proceso o recurso, son los que pueden ser medidos sólo con respecto a como el producto, recurso o proceso se relaciona con su entorno. Aquí el comportamiento del producto, proceso o recurso es importante, más en sí que la entidad.

2. La Medición en Ingeniería de Software

Entidades	Atributos	
Productos	Internos	Externos
Especificaciones	tamaño, reuso, modularidad, redundancia, funcionalidad...	facilidad de mantenimiento, comprensibilidad
Diseño	tamaño, reuso, modularidad, acoplamiento, funcionalidad	calidad, complejidad, facilidad de mantenimiento
Código	tamaño, reuso, modularidad, acoplamiento, funcionalidad, complejidad algorítmica, estructuración del flujo	confiabilidad, usabilidad, facilidad de mantenimiento
Datos de prueba	tamaño, nivel de cobertura	calidad
<i>Procesos</i>		
Especificación	tiempo, esfuerzo, número de cambios a los requerimientos	calidad, costo, estabilidad
Diseño detallado	tiempo, esfuerzo, número de errores encontrados en las especificaciones	costo, eficiencia de costos
Pruebas	tiempo, esfuerzo, número de errores encontrados en el código	costo, estabilidad, eficiencia de costos
<i>Recursos</i>		
Personal	edad, salario	productividad, experiencia, inteligencia
Equipos	tamaño, nivel de comunicación, estructuración	productividad, calidad
Software	precio, tamaño	usabilidad, confiabilidad
Hardware	precio, velocidad, tamaño de almacenamiento	confiabilidad
Oficinas	tamaño, temperatura, luz	comfort, calidad

Cuadro 2.1.: Ejemplos de componentes de la medición de software

2.4.1.1. Procesos

Frecuentemente tenemos preguntas acerca de nuestras actividades y procesos de desarrollo que las mediciones nos pueden ayudar a responder. Queremos saber cuanto tiempo toma completar un proceso, cuanto costará, si será efectivo o eficiente y como se compara con otros procesos que hemos elegido. Sin embargo, sólo una cantidad limitada de atributos internos de procesos se pueden medir directamente, entre ellos:

- La duración de un proceso o de una de sus actividades
- El esfuerzo asociado con el proceso o alguna de sus actividades
- El número de incidentes de un tipo específico que aparecieron durante el proceso o alguna de sus actividades

Muchas de estas medidas se usan en combinación con otras medidas para ganar un mejor entendimiento de que está sucediendo en un proyecto.

En algunos casos, desearíamos medir propiedades de un proceso que en sí mismo consiste de varios distintos procesos.

2.4.1.2. Productos

Los productos no deberían de ser restringidos a lo que la administración está comprometida a entregar al cliente, cualquier salida o documento producido durante el ciclo de vida del software puede ser medido y evaluado.

Atributos externos de los productos Hay muchos ejemplos de atributos externos de los productos. Dado que un atributo externo de un producto depende del comportamiento del producto y de su entorno, cada atributo medido debería de tomar estas características en consideración.

La usabilidad, integridad, eficiencia, reusabilidad, portabilidad, interoperabilidad y susceptibilidad de ser probado son otros atributos externos que podemos medir. Estos atributos describen no sólo el código sino también la otra documentación que apoya el esfuerzo de desarrollo. De hecho, la facilidad de mantenimiento, reusabilidad y hasta la susceptibilidad de ser probado de las especificaciones y diseños son tan importantes como el código en sí.

Atributos internos de los productos Los atributos internos de los productos algunas veces son fáciles de medir. Podemos determinar el tamaño de un producto al medir el número de páginas que llena o el número de palabras que contiene. Dado que los productos son concretos, tenemos un mejor entendimiento de algunos atributos como tamaño, esfuerzo y costo. Otros atributos internos de los

productos son más difíciles de medir, porque las opiniones de que significan y como se miden difieren. Hay muchos aspectos de complejidad de códigos y no hay un consenso claro de cómo medirlos.

Dado que los productos son relativamente fáciles de examinar de una forma automatizada, hay un conjunto de atributos internos comúnmente usados.

Los usuarios algunas veces ignoran como poco importantes estos atributos internos, basado en que como usuario el principal interés es la funcionalidad, calidad y utilidad del software. Aún así, los atributos internos pueden ser muy útiles al sugerir lo que es probable encontrar en los atributos externos.

La importancia de los atributos internos. Muchos métodos propuestos y desarrollados de ingeniería de software en los últimos 25 años dan reglas, herramientas y heurística para producir productos de software. Casi invariablemente, estos métodos dan una cierta estructura a los productos; se afirma que esta estructura los hace más fáciles de entender, analizar y probar. La estructura involucra dos aspectos del desarrollo:

- El proceso de desarrollo, porque ciertos productos necesitan hacerse en ciertas etapas
- Los productos en sí mismos, dado que los productos deben de satisfacer ciertos requisitos estructurales

En particular, la estructura de un producto está usualmente caracterizada por niveles de algunos atributos internos como modularidad, acoplamiento y cohesividad.

Atributos internos y el control de calidad Una buena razón por la cual los desarrolladores quieren usar atributos internos para predecir los externos es la necesidad de monitorear y controlar los productos durante el desarrollo.

Validación de medidas compuestas La “calidad” frecuentemente la usan los ingenieros de software para describir un atributo interno del diseño o el código. Aún así, la “calidad” es multi dimensional; no refleja un solo aspecto de un producto en particular.

Sin pensar en los atributos específicos que contribuyen a la noción general de calidad, muchas personas piensan que los muchas nociones cognitivas y estructurales como la complejidad, facilidad de mantenimiento y usabilidad se pueden capturar en un solo número como el producto interno bruto. Este número entonces debería de ser un indicador de todos los atributos que normalmente se asocian con los sistemas de alta calidad, como alta confiabilidad y alta facilidad de mantenimiento. Pero esta forma de abordar el problema ignora la pregunta de si los atributos internos que lo componen presentan una muestra completa y exacta del atributo más completo, y si la ponderación que se les asigna es adecuada.

2.4.1.3. Recursos

Los recursos que posiblemente se midan incluyen cualquier entrada de la producción de software. Por lo tanto el personal (personas o equipos), materiales (incluyendo papelería), herramientas (software y hardware) y métodos son candidatos para ser medidos. Se miden los recursos para determinar su magnitud, su costo y su calidad. Estas mediciones nos ayudan a entender y controlar el proceso al decirnos como el proceso está cambiando las entradas en salidas.

La productividad siempre es importante, y los gerentes de proyecto no sólo les interesa medirla, sino entender como mejorarla. Aunque sea una medida del personal, la productividad es una atributo externo, porque depende del proceso de desarrollo; un trabajador productivo en un proceso, podría hacerse menos productivo usando otro proceso.

La productividad normalmente se mide como

2. La Medición en Ingeniería de Software

$$\text{Productividad} = \frac{\text{Cantidad de salida}}{\text{Esfuerzo invertido}}$$

Hay que hacer notar que esta medida de recursos combina una medida de proceso (entrada) con una de producto (salida). La noción general es una noción de tipo económico, donde los negocios o mercados se juzgan comparándolos lo que entra de lo que sale. Para el desarrollo de software, la medida de salida usualmente se calcula como el número de líneas de código producidas como producto final, mientras que la medida de entrada es el número de personas mes usadas para especificar, diseñar, codificar, y probar el software. Aunque la analogía económica está incompleta para el software, dado que la cantidad de salida de software no está relacionada con la entrada de la misma forma que en la manufactura, dado que en la mayoría de los procesos de manufactura involucran la replicación de los productos, y el desarrollo de software es un proceso de creación, no de replicación, y la relación entre entradas y utilidades se define diferente.

Ningún desarrollador tiene el tiempo para medir y analizar todos los aspectos. Es importante enfocar las actividades de mediciones en aquellas áreas que necesitan la mayor visibilidad, entendimiento o mejora.

2.4.2. GQM como técnica para definir mediciones

Muchos programas de métricas comienzan con lo que es fácil o conveniente medir, en vez de medir lo que es necesario. Estos programas normalmente fracasan porque los datos resultantes no son útiles a los desarrolladores del software. Un programa de mediciones puede ser más exitoso si está diseñado teniendo en cuenta las metas del proyecto.

El paradigma GQM (Goal-Question-Metric) está enfocado hacia los procesos y sus mediciones. En el nivel superior del modelo de GQM están las metas de medición que se quieren cumplir, lo que constituye un “Qué quiero saber”. En el nivel intermedio están las preguntas que se utilizarán para responder esas metas de medición, lo que es un “Qué medir”. En el nivel más bajo, están las mediciones básicas que se utilizaran para poder darle un valor a las preguntas antes planteadas, lo que se podría pensar en un “Cómo medir”. Estos tres niveles están organizados en un árbol, de forma que una misma medición puede ser usada para contestar más de una pregunta, y una pregunta puede ser usada en la evaluación de más de una meta.

GQM tiene tres pasos:

1. Hacer una lista de las metas más importantes del proyecto de desarrollo o mantenimiento
2. Derivar de cada meta las preguntas que se deberán responder para determinar si las metas se están cumpliendo
3. Decidir qué deberá de medirse para poder responder las preguntas adecuadamente

Al derivar las mediciones de esta forma, queda claro como usar los datos resultantes.

De esta forma, sólo se generan aquellas mediciones que están relacionadas con la meta. En muchos casos, se necesitarán varias mediciones para responder una sola pregunta. También podría darse el caso de que una sola medición aplique a mas de una pregunta. La meta da la razón de juntar los datos, y las preguntas indican a los integrantes y al proyecto como usar los datos.

Lo que puede no ser evidente de un árbol o de una tabla de GQM, es que el modelo necesita combinar las mediciones en una forma coherente para que las preguntas se puedan responder. Por lo tanto, el enfoque de GQM se tiene que adicionar con un o más modelos que expresen las relaciones entre las métricas.

Aún cuando las métricas se expresen como una ecuación o una relación, la definición no siempre es clara o no ambigua. El árbol no dice como medir una línea de código o un punto de función, sólo que se necesita una medición del tamaño del código para contestar una pregunta acerca de la productividad. Se requiere trabajo adicional para definir cada métrica. En los casos que no existe una medición objetiva, se deberán buscar mediciones subjetivas.

En general, las metas típicas se expresan en términos de productividad, calidad, riesgo, satisfacción del cliente, etc, unido a verbos expresando la necesidad de evaluar, mejorar o entender. Es importante que las metas y las preguntas se entiendan en términos de su audiencia: una meta de productividad para un administrador de proyecto puede ser diferente que para un administrador departamental o para un director. Una medición en particular es útil sólo si ayuda a entender el proceso subyacente y uno o más de sus productos resultantes.

Una vez que hay una lista de posibles mediciones, hay que evaluar la organización para ver si es capaz de proveer información útil de las mediciones.

2.4.3. Medición y mejora de procesos

Las mediciones ofrecen visibilidad en las formas en las que los procesos, productos, recursos, métodos y tecnologías de desarrollo de software se relacionan unas con otras.

Las mediciones pueden ayudar a responder preguntas sobre la efectividad de técnicas o herramientas, la productividad de las actividades de desarrollo (como las pruebas o el control de configuración), la calidad de los productos, etc.

Adicionalmente las mediciones ayudan a definir una línea base para comprender la naturaleza y el impacto de los cambios propuestos. Finalmente, las mediciones ayudan a los gerentes de proyecto y desarrolladores a monitorear los efectos de las actividades y cambios en todos los aspectos del desarrollo, de forma que se puedan tomar acciones lo más pronto posible para controlar el resultado final si las mediciones que se realizan difieren significativamente del plan.

Las mediciones son útiles sólo en el contexto de la evaluación y la mejora. Elegir métricas, recolectar los datos, analizar los resultados y tomar acciones apropiadas requiere tiempo y recursos; estas actividades sólo tienen sentido si están dirigidas a metas específicas de mejora.

Algunos procesos de desarrollo son más naturales que otros, como lo hizo notar el SEI en su reporte de madurez de procesos y madurez de capacidades. Mientras algunas organizaciones tienen procesos bien definidos, otros son más variables, cambiando significativamente con la gente que trabaja en cada proyecto.

El SEI sugiere que hay cinco niveles de madurez de procesos; yendo de ad-hoc (el menos predecible y controlable), a repetible, definido, administrado y optimizando (el más predecible y controlable) y distingue entre dos de estos niveles en términos de actividades claves de los procesos que se realizan en cada nivel.

Pero todos estos modelos tienen una meta común, que usan la visibilidad en los procesos como un diferenciador clave entre un conjunto de “niveles de madurez”. Así, entre más visibilidad en el proceso completo de desarrollo, será más alta la madurez y habrá mayor entendimiento y control de las actividades de desarrollo y mantenimiento por los gerentes de proyecto y desarrolladores.

En cada nivel de madurez la medición y la visibilidad están muy relacionadas, un desarrollador sólo puede medir lo que esté visible en un proceso, y las mediciones ayudan a incrementar la visibilidad. Por lo tanto, una escala de madurez como la de cinco niveles que usa el SEI, es un contexto conveniente para determinar que medir primero y como planear un programa de mediciones que crezca para abarcar aspectos adicionales del desarrollo y administración.

Antes de identificar mediciones en particular que ayuden a responder estas preguntas, es útil determinar la madurez de procesos de la organización. Dado que la madurez de procesos sugiere que sea medido sólo lo que es visible, la incorporación de madurez de procesos con GQM da una idea más amplia de que medidas serán más útiles.

Las metas y el análisis de las preguntas son el mismo, pero de las métricas a usar varían con la madurez. Entre más maduro el proceso, más ricas pueden ser las mediciones. En otras palabras, entre más maduro el proceso, más maduras las mediciones.

Después de haber hablado de las mediciones desde el punto de vista de la Ingeniería de Software, en el siguiente capítulo daremos una introducción a un proceso de medición propuesto.

3. Proceso de Medición

El modelo de información de medición da un mecanismo formal para relacionar las necesidades de información con los productos y procesos de software. El plan de medición para un proyecto es una particularización del modelo de información que es específica para las necesidades de un proyecto. El modelo sirve como el recurso principal en el proceso de medición porque orienta toda la planeación y la implementación de la recolección de datos y de su análisis.

3.1. Plan de medición

La realización del plan de mediciones provee un método consistente para identificar las necesidades de información del proyecto, seleccionar y especificar las mediciones e integrarlas dentro de los procesos técnicos y administrativos del proyecto.

El objetivo de la actividad de planeación de medición es seleccionar y definir las mediciones que proveen la mejor visibilidad en las necesidades de información identificadas, debiéndose tratar primeramente las más prioritarias.

- La primera tarea es la identificación y la priorización de las necesidades de información. Éstas se definen explícitamente en el contexto de los objetivos, las características y las vulnerabilidades del proyecto.
- La segunda tarea es la selección y la especificación de mediciones apropiadas para especificar las necesidades de información identificadas. La tarea de seleccionar emplea un framework de experiencia que mapea necesidades identificadas de proyecto a mediciones específicas.
- La tercera y última tarea, involucra integrar las mediciones en los procesos del proyecto. Esto involucra examinar que tan útiles son las mediciones seleccionadas en el contexto de los procesos del proyecto, así como desarrollar procesos para recolectar y analizar los datos deseados en el entorno del proyecto. Los requerimientos de medición se deben usar para ganar visibilidad en los problemas relacionados a esos procesos y sus productos asociados.

Los resultados del proceso de planeación están documentados en un plan de medición de proyecto. El plan puede ser formal o informal, y puede estar incorporado en otro plan, como el plan de proyecto o el plan de desarrollo. Dada la naturaleza de los proyectos, puede surgir nuevas necesidades de información o cambiar las existentes en el curso del proyecto, se pueden proponer mediciones alternas para satisfacer mejor las necesidades o para minimizar los costos de medición, o las prioridades previamente definidas pueden cambiar. El plan de medición de proyecto es un documento vivo en constante evolución.

3.1.1. Identificar las necesidades de información

Todos los proyectos de software tienen ciertos objetivos, que pueden ser establecidos por la dirección o negociados por el líder de proyecto con el cliente. Estos objetivos se definen normalmente en términos de funcionalidad de sistema, recursos presupuestales, hitos, niveles de calidad y metas de productividad del sistema o de la empresa. Existen áreas vulnerables que pueden impactar la realización de los objetivos del proyecto, estos pueden ser riesgos, problemas y falta de información.

3. Proceso de Medición

Se deben de considerar múltiples fuentes de información para asegurar que se identifican todas las necesidades de información. Algunas fuentes pueden ser:

- **Evaluaciones de riesgos**, pueden apuntar a necesidades de información relacionadas a los requerimientos, la tecnología, los procesos, costos o calendario.
- **Supuestos y restricciones del proyecto**, el plan de proyecto usualmente está basado en muchos supuestos. La falta de información debe ser tratada como una necesidad de información porque puede impactar los estimados de esfuerzo, calendario, y calidad.
- **Tecnologías usadas**, si los objetivos del proyecto dependen de tecnologías específicas, la efectividad de esas tecnologías es una necesidad de información.
- **Criterios de aceptación de producto**, los clientes pueden poner metas o criterios estrictos para la entrega de un producto de software. Si hay una duda razonable de la habilidad de la organización para cumplir los criterios de aceptación, los objetivos u otros criterios externos predefinidos, se debe identificar esos criterios como una necesidad de información.
- **Requerimientos externos**, algunas necesidades de información pueden estar relacionadas a requerimientos y preocupaciones externos al proyecto. La probabilidad de cumplir metas organizacionales agresivas o irreales, debe ser considerado como una necesidad de información.
- **Experiencia**, un equipo de proyectos con experiencia en proyectos similares debe poder identificar áreas problema como necesidades de información.

Cada necesidad de información debería de estar especificada en la terminología de proyectos adecuada. El énfasis debe de estar en aquellos aspectos de las necesidades de información más importantes para el proyecto.

Un proceso de identificación conjunta, con representación de todos los involucrados (stakeholders), es una forma de extraer necesidades de información y alcanzar un consenso en las prioridades de medición; una amplia participación también promueve el compromiso con el proceso de medición.

Las mediciones y la administración del riesgo son sinérgicos. Ambas disciplinas enfatizan la prevención y la detección temprana de problemas en vez de esperar a que los problemas se vuelvan críticos. El proceso de medición tiene un papel en la administración de riesgos al indicar cuando los riesgos se están constituyendo en problemas y si los planes de mitigación están teniendo el efecto deseado.

3.1.2. Priorizar las necesidades de información

Los gerentes de sistemas de software típicamente tienen muchas necesidades de información. Éstas deben estar priorizadas para asegurarse que el programa de medición cubra con las necesidades de información que más benefician a los objetivos del proyecto.

Hay muchas formas de establecer prioridades. Usar criterios bien establecidos permite alcanzar consensos de las prioridades entre los participantes del proyecto. Las necesidades de información que tengan más notoriedad, normalmente relacionadas con calendarios agresivos, restricciones presupuestales y cambios en los requerimientos, deberán de ser atendidas primero.

El enfoque de priorización siempre reflejará cierto grado de subjetividad, la cual deberá ser minimizada. No importando como se prioricen las necesidades de información, se deberán agrupar por similitud. Esto hace que sea más fácil detectar mediciones que sean útiles para satisfacer múltiples necesidades de información.

La priorización de la información de proyecto es dinámica. Una vez que el proyecto esté andando se podría identificar información adicional, y los riesgos y su probabilidad cambian al avanzar el proyecto y acercarse las metas y entregas. Por lo tanto el proceso de medición tiene que cambiar para mantenerse al día.

3.1.3. Describir el contexto del proyecto

La definición de una medición no depende sólo de una necesidad definida de información. Los procesos técnicos y administrativos también deben ser tomados en cuenta. El definir un programa de mediciones requiere un entendimiento del contexto en el cual las mediciones se llevarán a cabo. Tiene poco sentido definir mediciones que no se pueden implementar en un proceso existente.

Algunos factores clave a considerar son:

- El modelo de ciclo de vida o actividades de estructura usadas para definir el proceso de software del proveedor.
- La estructura del producto, incluyendo incrementos y la asignación de varias tareas a terceros externos.
- Las actividades de medición actuales.
- La tecnología de software y de sistema, incluyendo técnicas de diseño, lenguajes de programación, entornos y herramientas.
- Las fuentes planeadas de componentes de software, incluyendo los comprados, desarrollados recientemente y componentes reusados.
- Las prácticas de administración, coordinación, revisión, prueba y revisión.
- Los estándares de ingeniería y administración que se aplicarán.
- La madurez de procesos de cada organización involucrada en el proyecto.

Cuando sea posible, se deberían de usar los procesos y mecanismos de recolección de datos existentes, así como minimizar los requerimientos nuevos de medición. Se puede tomar como base el WBS (work break down structure) para recolectar y agrupar los datos.

3.1.4. Seleccionar las mediciones aplicables

El siguiente paso al seleccionar y especificar las mediciones asociadas es elegir la mejor combinación entre las mediciones base y derivadas asociadas con cada categoría de medición. Hay que considerar la concordancia entre las mediciones potenciales y las necesidades de información, así como la efectividad y eficiencia de las mediciones en sí y la habilidad del equipo de proyecto de producirlas.

Con los siguientes criterios de selección se pueden tratar de identificar las mejores mediciones para un proyecto.

- **Efectividad de las mediciones**, ¿qué tan efectiva en proveer la visibilidad deseada es la medición?, ¿es una medición directa de la característica proceso o producto en cuestión?, ¿la medición es útil para más de una necesidad de información?
- **Características del dominio**, ¿hay ciertas mediciones más adecuadas en un cierto dominio de aplicación?
- **Prácticas de administración de proyectos**, considerar si se pueden utilizar las prácticas actuales de administración de proyectos.
- **Costo y disponibilidad**, qué datos estarán disponibles inmediatamente en el contexto del proyecto. La recolección electrónica de datos cuesta menos que la recolección manual
- **Cobertura del ciclo de vida**, ¿la medición aplica a la fase que se va a considerar? ¿aplica a múltiples fases?

3. Proceso de Medición

- **Requerimientos externos**, ¿la organización tiene algunos requerimientos de medición?
- **Tamaño y origen de los componentes del sistema**, ¿el tamaño del proyecto justifica una mayor inversión en medición? ¿alguna medición tiene sentido si mucho del sistema es comprado, reuso de software o desarrollado externamente?

Muchas mediciones diferentes pueden aplicar a una misma necesidad de información, pero por lo general, no es práctico recolectar todas o la mayoría de las mediciones potenciales. Así mismo, en la mayoría de los casos el proceso de selección requiere de ponderaciones (trade-off) entre los criterios de selección de mediciones.

3.1.5. Especificar los indicadores

Una vez que se han seleccionado los candidatos a medir, se deben definir las medidas básicas, derivadas y los indicadores para completar la especificación de los indicadores. Esta especificación provee una definición operativa de las medidas relevantes.

Las especificaciones deben estar documentadas para ser fácilmente referibles, usualmente como parte del plan de medición. Éstas pueden variar conforme avanza el proyecto al irse modificando y actualizando los procesos. Mantener una base de experiencia de mediciones anteriores (guardados en forma de plantillas llenas) también puede facilitar la planeación.

Un indicador es una medida que provee una estimación o una evaluación de unos atributos derivados de un modelo de análisis con respecto a una necesidad de información. Combinar mediciones base o derivadas usando un algoritmo o un modelo predeterminado genera un indicador. Típicamente el valor de un indicador es un número o una serie de números y se puede representar en una gráfica o una tabla. Varias medidas base y derivadas se pueden combinar de diversas maneras para construir indicadores.

Un buen indicador:

- Soporta el análisis de la necesidad de información propuesta
- Soportar el tipo de análisis necesarios (estimación, factibilidad, análisis de rendimiento)
- Proveer el nivel adecuado de detalle
- Indicar una posible acción de la gerencia
- Proveer información oportuna para tomar decisiones y acciones

Los indicadores normalmente comparan valores medidos con una expectativa de lo que deberían ser. La expectativa puede estar basada en promedios históricos, valores planeados, límites específicos o valores umbral. El uso de modelos y criterios de decisión ayuda a decidir si las diferencias entre los datos esperados y actuales es significativa.

Desarrollar y disseminar claramente las especificaciones de todas las mediciones facilita la recolección de datos consistentes y verídicos.

3.1.6. Integrar el enfoque de medición en los procesos del proyecto

Durante la planeación de la medición, identificar y explotar cualquier mecanismo de medición existente debería de ser una prioridad, sobre todo al implementar mediciones en un proyecto en curso. El uso de fuentes de datos existentes aprovecha la familiaridad y potencialmente puede disminuir el costo de implementar el programa de medición.

Los datos de medición pueden venir de muchas fuentes simultaneas. Extraer y entregar esos datos de fuentes electrónicas normalmente es más barato que los métodos manuales de recolección.

Las mediciones, pueden típicamente originarse de estas fuentes:

- Datos históricos, incluye los datos recolectados de proyectos anteriores
- Datos de planeación, típicamente representa los valores estimados y esperados.
- Datos de desempeño del proyecto, conforme el proyecto avanza, comienza a haber disponibilidad de datos. Existen muchas fuentes de datos en los procesos técnicos y administrativos.

Para necesidades de información de alta prioridad, se deberían de considerar fuentes de datos que estén disponibles desde etapas tempranas del proyecto.

Establecer unas buenas especificaciones es tan importante como usar los datos existentes al implementar las nuevas mediciones.

3.1.7. Desarrollar procedimientos de medición

Se deberán de desarrollar procedimientos tanto para la recolección de datos como para su almacenamiento, análisis y reportes.

Estructuras de agregación Al medir un componente de software también puede proveer información acerca del producto al cual pertenece, si se entiende la relación. Los procedimientos de agregación capturan las relaciones entre las entidades de las mediciones para que los datos se puedan combinar y descomponer, lo que sea más apropiado para el caso.

Las estructuras de agregación pueden ser por:

- Estructura de componentes, la organización del software en subsistemas, componentes, unidades, etc; da una base para agregar información acerca del tamaño y calidad.
- Estructura funcional, la descomposición de los requerimientos en funciones provee una base para agregar la información de calidad y tamaño funcional.
- Estructura de actividad, la organización de los procesos de trabajo en elementos y tareas se presta para agregar el esfuerzo y otros datos de los recursos.

Cada entidad puede ser mapeada en una o más estructuras de agregación, de forma que los datos se puedan combinar apropiadamente durante el análisis.

Periodicidad La recolección, el análisis y los reportes de los datos necesarios necesitan ser realizados con una frecuencia suficiente para soportar el proceso de toma de decisiones que generó la necesidad de información.

Sin embargo, es posible que sea necesario recolectar datos mucho más frecuentemente para garantizar la calidad de los datos. Los recuerdos humanos se pueden deteriorar o perder, incluso durante el periodo de un mes. Aunque la mayoría de los análisis y reportes se hacen mensualmente, los problemas, se podrían analizar semanalmente. También existe el caso de información que sólo se recolecta ocasionalmente, como al terminar una fase de proyecto.

Asignación de la responsabilidad a personas La responsabilidad de la realización de cada paso en el proceso de medición, se debería de asignar a las personas apropiadas.

Fuente Se deberá especificar la fase y la actividad durante la cual se recolectó y analizó cada medición base.

3. Proceso de Medición

Mecanismos de análisis y reportes Las herramientas y procesos para analizar mediciones base y mediciones derivadas para producir los indicadores se debería identificar. Los análisis pueden ser tan difíciles y costosos como la recolección de datos. Incluso con datos disponibles, la falta de una herramienta para analizarlos, puede hacer demasiado caro el uso de los datos.

Planeado vs actual Muchos análisis involucran la comparación del desempeño actual contra lo planeado. Muchas medidas incluirán una estimado o planeado y un resultado actual. Las actualizaciones a los planes y estimados deberán también de actualizarse en los datos.

Control de configuración La configuración de los productos intermedios y finales de la medición debería de controlarse de la misma forma que el código, al menos asociar las fechas de recolección y análisis con los datos y los reportes.

3.1.8. Documentar el plan de medición del proyecto

La última tarea del proceso de planeación es documentar el plan que deberá de contener todos los resultados incluyendo información de las necesidades, mediciones y procesos de medición. El plan deberá explicar en suficiente nivel de detalle cómo se integrará el proceso de medición en el proceso de administración de proyectos de forma que los tomadores de decisiones puedan usar los resultados de la medición; así mismo el plan de medición deberá de estar coordinado con el plan de administración de riesgo.

Un plan bien desarrollado asegura que:

- El proceso de medición es repetible.
- Los orígenes de datos, los analistas y los tomadores de decisiones comprenden las mediciones, los análisis asociados y las recomendaciones.
- Se pueden hacer comparaciones entre los niveles de proyecto y organizacional.
- El proceso de medición puede continuar, aún con cambios de personal.

3.2. Realizar la medición

El plan de medición provee la guía para implementar la medición en un proyecto. La primera tarea en la medición del desempeño incluye implementar los procedimientos de recolección de datos para recabar las mediciones base definidas en el plan. A continuación, se ejecutan los procedimientos de análisis de datos para producir los indicadores necesarios para realizar el análisis planeado. Finalmente, la información producida en el análisis se usa para formular recomendaciones que se le reportan a los tomadores de decisiones, junto con los resultados de los análisis.

3.2.1. Recolectar y procesar los datos

Esta tarea involucra recolectar datos de varias fuentes identificadas en el plan de medición, preparándolas para el análisis, y luego guardarlas para que estén disponibles para ser analizadas.

Las tres tareas básicas son:

- Cómo poner disponibles los datos, y capturarlos adecuadamente
- Cómo asegurar la calidad de los datos
- Cómo almacenar y administrar los datos para el análisis.

Como se capturan los datos durante las actividades del proyecto afecta el costo y la exactitud, en muchos casos los datos se pueden capturar automáticamente; en otros se necesitarán capturar por procesos manuales.

También es importante como se almacenan los datos una vez que se generan: una forma accesible electrónicamente es lo más deseable.

La confiabilidad de un proceso flexible de análisis depende en la efectividad de la administración de los datos. Como otro activo valioso del proyecto, los valores de las mediciones y los indicadores deberá ser administrado y controlado. Todos los datos deberán de estar identificados con su fecha de recolección y fuente, lo que ayudará a alinear los hallazgos y sus conclusiones con los eventos del proyecto, así como permitir comparaciones de diferentes periodos históricos.

Los datos deberán de ser verificados antes de ser analizados. La verificación de datos deberá de considerar tanto la exactitud de los datos guardados, como la fidelidad con la que son procesados. La verificación de datos requiere especificaciones precisas de las mediciones y un entendimiento de los procesos del proyecto.

3.2.2. Analizar los datos

El objetivo principal de esta tarea es el análisis de los datos recolectados con anterioridad. Es aquí donde se enfocará mucho del esfuerzo de medición. Esta tarea transforma los valores de las mediciones base en valores para indicadores. Los indicadores y sus criterios de decisión asociados se usan para tomar decisiones de planeación o determinar la necesidad de tomar acciones correctivas una vez que el proyecto ha comenzado. Estos análisis se realizan al ejecutar los procedimientos definidos en el plan de medición, sin embargo, como no se pueden anticipar todas las necesidades ni todas las preguntas de la gerencia durante la creación del plan, el procedimiento de análisis de datos debe también debe de contemplar el uso de técnicas alternas de análisis.

Los tres tipos de análisis empleados en un proyecto para soportar las decisiones de administración de proyectos son: estimación, análisis de factibilidad y análisis de desempeño. Cada tipo de análisis requiere sus propias técnicas de medición, tiene sus propias entradas y produce diferentes tipos de resultados.

3.2.3. Generación de indicadores

Pocas necesidades de información se pueden satisfacer recolectando sólo valores de desempeño actuales. Los datos actuales deben de compararse con una expectativa de lo que deben de ser. Estos valores esperados usualmente se convierten en las medidas básicas de un indicador, adicionalmente, se usan criterios decisión predefinidos para decidir si las diferencias entre los datos actuales y los esperados es significativa.

3.2.4. Representación de los indicadores

Los indicadores normalmente se despliegan mejor con una gráfica que con una tabla de valores. Las gráficas hacen más aparentes y fáciles de interpretar las tendencias de datos, varianzas y relaciones.

Las gráficas bien diseñadas facilitan la comunicación de los resultados de la medición. Las gráficas no deben ser complejas, y deben de comunicar claramente un mensaje. Es mejor utilizar múltiples gráficas que una con muchas mediciones.

3.3. Técnicas de análisis

3.3.1. Estimación

La estimación produce proyecciones cuantitativas de atributos claves del proyecto. Estas proyecciones forman la base para los planes iniciales del proyecto.

La estimación normalmente es el primer tipo de análisis que se realiza en el proyecto. La primera ronda de estimación ocurre muy al principio del proyecto, incluso antes de que se elija el equipo para el proyecto; aquí se establecen los recursos y costos gruesos así como compromisos de entrega. Dado que los estimados iniciales son poco exactos, deben actualizarse conforme avanza el proyecto, y estos nuevos estimados retroalimentarse al plan de proyecto.

Estimaciones mal hechas e ideas equivocadas acerca del proceso de estimación frecuentemente son un factor importante de fracaso en proyectos. En muchos casos las estimaciones mal hechas llevan a planear metas que son imposibles de cumplir, lo que produce retrasos, bajo desempeño y baja calidad [McC06]. Algunas de las causas son:

- Falta de experiencia en la estimación
- Falta de datos históricos en los cuales basar los estimados
- Falta de un proceso sistemático de estimación, técnicas probadas o modelos adecuados a las necesidades del proyecto.
- No incluir en el estimado actividades esenciales del proyecto o del producto.
- Expectativas o supuestos irreales
- Falta para reconocer y tomar en cuenta la incertidumbre inherente en los estimados de proyecto

3.3.1.1. Proceso de estimación

El primer paso en el proceso de estimación es seleccionar, de las muchas disponibles, la técnica de estimar. El segundo paso es mapear la técnica a la secuencia de actividades del proyecto y del producto y calibrar los modelos asociados con datos históricos de la organización.

El tercer paso es calcular los estimados de tamaño, esfuerzo, calendario y/o calidad usando la técnica. El cuarto paso es evaluar el estimado y determinar si se necesita ajustar para alinearlo a las restricciones de proyecto.

3.3.2. Análisis de factibilidad

El análisis de factibilidad evalúa la exactitud y el realismo de los planes de proyecto. Para que un plan de proyecto sea factible, los elementos individuales del plan deben ser realistas y alcanzables, y consistentes en sus relaciones unos con otros.

Cuando hay un problema de factibilidad, no necesariamente todo el proyecto es no factible, sino sólo partes de él.

3.3.2.1. Indicadores de factibilidad

La factibilidad de las medidas del plan (baseline), son la prioridad en este tipo de análisis. Éstas se comparan típicamente con otras medidas planeadas para evaluar el realismo y la consistencia interna de los planes de proyecto. Para evaluar la factibilidad de los planes se pueden usar datos históricos, umbrales o normas establecidas.

Indicadores basados en tendencias Los planes pueden representarse como tendencias en el tiempo. Las tendencias se representan de 2 formas: como un plan que muestra la cantidad total acumulada que debe ser alcanzada hasta la fecha como el costo total hasta la fecha.

La otra forma es un plan que muestra la cantidad asignada al proyecto por periodo, como por mes.

Indicadores basados en umbrales o metas Algunas veces el baseline es un único valor, como una meta, límite o umbral contra el cual los valores se compararán. Normalmente se originan de los requerimientos, como complejidad de código, tiempo de respuesta o utilización de memoria, o de los criterios de aceptación de producto.

3.3.2.2. Proceso de análisis de factibilidad

Las entradas son los datos planeados y sus estimados, así como cualquier restricción externa, para generar indicadores. Si los planes no son alcanzables o son poco realistas se debe identificar como un riesgo, y se debe de trabajar en caracterizar y evaluar la magnitud del riesgo. Se debe localizar el origen del riesgo para evaluar su impacto. Se pueden generar más indicadores para ayudar a cuantificar el riesgo de proseguir con los planes como están; una vez que se entiende la magnitud del problema, se deberán discutir alternativas, revisiones al plan, estrategias de desarrollo y formas de mitigarlo.

3.3.3. Análisis de desempeño (performance)

El análisis de desempeño determina si el proyecto está cumpliendo sus planes y metas. Independientemente de su factibilidad, una vez que un proyecto se ha comprometido a un plan, el desempeño se puede medir contra el plan. El administrador de proyecto debe de prestar atención en la adherencia que tiene el proyecto al plan. Aún cuando el proyecto comience con un buen plan, si el desempeño comienza a desviarse del plan, se deberá identificar las razones de la desviación y tomar acción correctiva. La meta del análisis de desempeño es dar a los tomadores de decisiones información a tiempo para afectar el resultado del proyecto. El análisis de desempeño deberá de ser realizado regularmente durante la duración del proyecto.

El proceso debe de ser flexible para poder analizar los sucesos durante el proyecto, y deberá de verse como una actividad investigativa en la que el análisis de las mediciones se use para identificar y aislar los problemas, lo cual puede necesitar el uso de datos diferentes, la generación de indicadores diferentes y la identificación de cursos alternos de acción cada vez que se realice el análisis.

3.3.3.1. Indicadores de desempeño

Los mismos tipos de indicadores con los cuales se monitorea la factibilidad se usan para monitorear el desempeño, los basados en tendencias o en metas. La diferencia entre los dos es si el plan es relativamente constante o cambia conforme avanza el tiempo.

Indicadores basados en tendencias Los indicadores basados en tendencias se usan cuando los valores esperados o planeados cambian regularmente en el tiempo. El análisis de desempeño de un indicador de tendencia determina si el desempeño actual corresponde al planeado.

Indicadores basados en umbrales o metas Los indicadores de desempeño basados en umbrales o metas se usan cuando los valores esperados permanecen constantemente relativamente en el tiempo. El análisis de desempeño determina si el proyecto actual cumple o excede estos límites establecidos, o mientras se mantiene dentro de los límites establecidos. Se debería de investigar cuando los valores actuales salen del rango.

Fijar los umbrales en indicadores basados en tendencias

3. Proceso de Medición

Una técnica común para monitorear el desempeño es fijar umbrales basados en el desempeño planeado. Mientras que el proyecto progresa de acuerdo al plan, no se requerirá acción del gerente. Los umbrales se pueden mostrar gráficamente como fronteras en las líneas de tendencia.

Hay tres técnicas para fijar los umbrales:

Arbitrarios, usualmente un porcentaje basado en juicio. Es el menos deseable, pero el más fácil de poner. Usualmente representan el punto a partir del cual tomará acciones.

Reserva del gerente, la reserva es la cantidad de recursos no asignados que pueden ser usados para hacer trabajo restante. Una reserva puede ser planeada, como un búfer en el plan de proyecto. Este tipo de umbral se va acortando conforme se va consumiendo.

Límites estadísticos, en esta técnica, se usan datos reales para calcular estadísticamente un umbral. Pueden ser límites o intervalos de confianza basados en datos estadísticos que representan la probabilidad de un resultado de caer a cierta distancia del promedio. Son los límites más difíciles de poner porque requieren datos de buena calidad durante un periodo de tiempo más largo, pero tienen como ventaja quitar la subjetividad de los umbrales arbitrarios.

Los umbrales son pistas para alertar a la alta dirección a investigar y posiblemente tomar acción.

3.3.3.2. Proceso de análisis de desempeño

El análisis de desempeño consiste de cuatro pasos. El primero evalúa el desempeño actual contra el plan, se generalizan los indicadores y se analizan. Si se identifican problemas o riesgos, se pueden realizar los siguientes pasos.

El segundo paso es evaluar el impacto de los problemas, localizar las fuentes de problemas y cuantificar el tamaño del problema. Es posible que se necesite generar indicadores adicionales en este paso.

A continuación se predice el resultado del proyecto, posiblemente extrapolando las tendencias actuales de los datos. Finalmente si el resultado obtenido no cumple con los objetivos del proyecto, se deberán de identificar y evaluar acciones alternas, y esta información se le da al administrador de proyectos como una entrada para la toma de decisiones.

Varias de las tareas de análisis de desempeño requieren de información no medible. Las decisiones no se pueden basar solamente en información cuantitativa. Se deberá de considerar el contexto del proyecto desde el punto de vista de los clientes, desarrolladores, revisiones técnicas y gerenciales, revisiones de documentos y análisis de riesgos. Reunir e integrar información no cuantitativa es esencial para un programa de medición exitoso.

3.3.4. Hacer recomendaciones

El propósito último de la medición es ayudar a los ingenieros, clientes y gerentes a tomar decisiones más informadas, por lo tanto la última tarea de la realización de mediciones es hacer recomendaciones y comunicarlas junto con los análisis de los resultados a los tomadores de decisiones adecuados.

- El análisis de las vulnerabilidades del proyecto.
- Evaluación global del proyecto, deberá incluir su status usando las necesidades de información del proyecto y proyecciones de desempeño hasta que se complete.
- Identificación de problemas, riesgos y falta de información específica.
- Recomendaciones, como acciones alternas, con ventajas y desventajas de cada una, para atacar los problemas identificados en el análisis.

- Nuevos problemas potenciales, la naturaleza del proceso de medición o de las acciones propuestas puede dar origen a nuevas necesidades de información, que pueden afectar el enfoque del proceso de medición.

Los resultados de la medición deberán poderse incorporar en los reportes de status de los proyectos que se da a la alta dirección y a los clientes. El análisis ayuda a explicar cómo diferentes factores contribuyen a ciertos resultados del proyecto, esto a su vez ayuda a dirigir a los tomadores de decisiones hacia una discusión productiva y no especulativa de alternativas cuando se necesita acciones correctivas.

Se deben de tomar acciones para tener cualquier beneficio de las mediciones. Las recomendaciones bien comunicadas facilitan la acción.

La medición es más fácilmente aceptada y usada cuando los resultados son fácilmente accesibles. Las juntas de equipo de proyecto proveen un foro común para revisar los indicadores, sacar conclusiones y formular recomendaciones, este tipo de interacción en equipo puede descubrir eventos e información cualitativa que ayuda a explicar los datos.

Los sistemas web son otra forma cada vez más popular para difundir los resultados de las mediciones [MCJ+02].

Tras haber hablado de los procesos de medición en este capítulo, hablaremos de los sistemas de soporte a la toma de decisiones en el siguiente.

4. El apoyo a la toma de decisiones

El concepto de sistemas de soporte a decisiones (Decisión Support Systems, DSS) está centrado en el paradigma de soporte. Esto es, una computadora es puesta al alcance del tomador de decisiones, que podrá usar datos o modelos para reconocer, comprender y formular un problema, y usar ayudas analíticas para evaluar alternativas.

4.1. Los objetivos de DSS

Un DSS puede definirse como:

Un sistema computacional que provee información de un dominio específico usando modelos analíticos de decisión y acceso a bases de datos, para ayudar a una persona que toma decisiones a hacer decisiones efectivas en tareas complejas y poco estructuradas. [KM90]

Los DSS son útiles cuando existe una meta fija pero no hay una solución algorítmica. Los caminos de solución pueden ser numerosos y dependientes del usuario. En otras palabras, la meta de un DSS es mejorar una decisión al comprender mejor y preparar mejor las tareas que llevan a la evaluación y elección (colectivamente llamado el proceso de decisión).

4.2. Características de los problemas poco estructurados

Los problemas estructurados son rutinarios y repetitivos, porque no son ambiguos (dado que cada problema tiene un solo método de solución). Un problema menos estructurado puede tener más de un método alternativo de solución, y las soluciones podrían no ser equivalentes. Un problema completamente “no estructurado” tiene un método de solución desconocido o demasiadas soluciones para evaluar eficientemente.

Son problemas de decisiones poco estructurados los que las preferencias, juicios, intuición y experiencia de la persona que toma las decisiones que son esenciales, y son los que más se benefician del enfoque de DSS.

- La búsqueda de la solución implica una mezcla de:
 - Búsqueda de información.
 - Formalización, o definición y estructuración del problema (modelado del sistema).
 - Procesamiento.
 - Manipulación de datos.
- La secuencia de las operaciones precedentes no se conoce de antemano dado que:
 - Puede ser en función de los datos.
 - Puede ser modificado, dado resultados parciales.
 - Puede estar en función de preferencias de usuarios.
- Los criterios de la decisión son muchos, en conflicto y altamente dependientes de la percepción del usuario.

4. *El apoyo a la toma de decisiones*

- La solución deberá ser obtenida en un tiempo limitado.
- El problema evoluciona rápidamente.

4.3. **Estrategia de evolución de un DSS**

La estrategia de implementación de un DSS está enfocada de forma que inicialice el proceso de aprendizaje de los usuarios y lo oriente hacia el proceso evolutivo del propio DSS.

1. El diseñador diseña e implementa un entorno de desarrollo de DSS.
2. El usuario aprende cómo usar el entorno de desarrollo.
3. El usuario (o usuario+diseñador) implementa el primer DSS o mejora el existente.
4. El usuario trabaja con el DSS para realizar sus tareas y resolver problemas.
5. El usuario identifica nuevos problemas o deficiencias en su forma actual de resolver problemas.
6. Si el usuario puede extender el DSS existente añadiendo nuevos datos, nuevos conceptos, nuevos modelos y nuevas presentaciones, vuelve a comenzar del paso 3 al 7.
7. El diseñador extiende las capacidades del entorno de desarrollo de DSS añadiendo un subsistema de DBMS, extendiendo los conceptos del lenguaje de modelado, extendiendo las posibilidades del generador de gráficas, añadiendo nuevos datos y nuevos subsistemas.

4.4. **Mejoras esperadas del uso de un DSS**

Las mejoras por el uso de un DSS que han sido publicitadas se pueden resumir en los siguientes apartados.

4.4.1. **Mejorar la efectividad de la toma de decisiones**

Se puede decir que la mejor calidad de las decisiones que se reporta en la literatura al usar un DSS es por las siguientes razones:

- Mayor acceso a la información
- Reconocimiento del problema más rápido y eficiente
- Mayor acceso a herramientas computacionales y modelos probados para calcular criterios de decisión
- Habilidad para generar y evaluar grandes números de alternativas

4.4.2. **Mejora de la eficiencia de la toma de decisiones**

Los tres componentes principales con respecto a la eficiencia de usar un DSS son:

- Reducción del costo de la decisión
- Reducción del retraso para alcanzar una decisión para el mismo nivel de detalle en el análisis
- Mejor calidad en los documentos impresos que soportan la decisión

4.4.3. Mejorar la comunicación y colaboración entre las personas que toman las decisiones

La diferencia entre los niveles de conocimiento tiende a desaparecer dado que las variables y las relaciones se hacen explícitas y sus valores y naturaleza se pueden discutir más fácilmente y con precisión. Las divergencias entre las personas que toman las decisiones se mantienen, pero entonces existe un conocimiento preciso de la naturaleza de la diferencia de opinión, y la forma en a la que se llegó a cada alternativa. También, compartir un modelo ayuda a separar las diferencias entre hechos y preferencias personales.

Se reconoce que ciertas personas podrían no estar interesadas en hacer explícitos los factores de una decisión, y como consecuencia, no les parecerá usar un DSS. Pero usualmente el problema es porque estas personas no están interesadas en la mejora de la decisión o la discusión de sus motivaciones y preferencias reales. La decisión ya esta tomada (tal vez usando el DSS) y su meta es que se adopte la alternativa que ellos proponen, por lo tanto, esto es otro problema: ¿cómo hacer que otros adopten o compartan tus preferencias?

4.4.4. Mejorar el proceso de aprendizaje de los usuarios

Esta consecuencia de usar un DSS está claramente relacionada con la de la efectividad. Dicho de otra forma, dado que un DSS mejora el proceso de aprendizaje de los usuarios la efectividad de las decisiones mejora. Esta consecuencias ha sido considerada incluso como un criterio de diseño: un DSS bien diseñado es uno que agiliza el proceso de aprendizaje de sus usuarios.

Se asume que la habilidad de resolver problemas recae en dos componentes principales: conocimiento (en particular, conocimiento metodológico) y experiencia, entonces el ambiente proveerá una forma fácil de usar ese conocimiento, y también una forma de aumentar la velocidad de acumular experiencia, dado que es mucho más fácil estudiar casos.

4.5. Posibles problemas derivados del uso de un DSS

Se debe también resaltar el hecho que la disponibilidad de un DSS bien diseñado, puede provocar algunos efectos adversos. Se supone que los DSS están para ayudar a los equipos a tomar decisiones al dotarlos de un mejor entendimiento de las consecuencias de sus decisiones y como se relacionan con sus preferencias. Pero la tarea principal de un manager es actuar en su entorno para tratar de alcanzar las metas de su organización que provee un producto o servicio a su entorno (clientes y usuarios). El gerente debería de estar conciente permanentemente de no olvidar tratar de propiciar un balance entre estudiar el problema y actuar sobre el entorno. Usar un DSS, por muy bueno que sea, no cambia el entorno, sino que los entornos son cambiados por personas que actúan sobre ellos.

4.5.1. Restricción de las elecciones

El usuario deberá diagnosticar si el DSS le permite extender el tipo de decisiones para las cuales fue producido (y bajo el tiempo y presupuesto disponible) para incluir una nueva decisión. Si no se puede hacer esta extensión sería mejor no usar el sistema.

4.5.2. Asumir relevancia

Una vez que se ha instalado un sistema de cómputo es difícil evitar asumir que las cosas con las que se puede lidiar son las cosas más relevantes para el manager. Una aplicación computacional se construye bajo una serie de supuestos. Si el usuario no tiene la habilidad de reconocer que de las condiciones de uso del sistema no encajan en estos supuestos, entonces el sistema puede ser pernicioso. El problema es que el diseñador podría no estar suficientemente conciente de éstas hipótesis.

4.5.3. Transferencia de poder no intencional

El problema normalmente se origina si el diseño hecho por especialistas en cómputo tiende a poner más énfasis en el valor de la eficiencia y a devaluar la “necesidad de criterio e innovación humana”. Este argumento es más válido en el caso de una total automatización del proceso de decisión y mucho menos notorio para un DSS dado que el proceso de decisión sólo es asistir.

4.5.4. Obscurecimiento de la responsabilidad

Una vez que un sistema de cómputo se diseña y se pone en producción, tiende a ser tratado como una entidad independiente. La computadora deja de ser un “experto secundario” y se convierte en un intermediario, un medio sofisticado de comunicación.

Debemos ser cautos de recordar que la computadora es un intermediario. Se puede, por supuesto, sólo coincidir con la idea que el DSS es un medio sofisticado de comunicación, y la responsabilidad no puede ser sólo de la máquina; sin embargo, es esencial asegurarse que los futuros usuarios están adecuadamente instruidos para recordar que cualquier DSS se construyó sobre una serie de supuestos acerca de cómo se usaría el sistema y como sus respuestas se interpretarían. Siempre hay un riesgo que un DSS se use de formas que no encajen con los supuestos del experto o del diseñador; el único remedio es la educación.

Tras esta introducción a los sistemas de toma de decisiones, en el siguiente capítulo hablaremos de la administración y el reuso de la experiencia.

5. La administración y el reuso de la experiencia

Muchas de las soluciones a los problemas complejos dependen de la experiencia. El compartir la experiencia permite tener acceso a la experiencia de otras personas y evitar cometer los mismos errores que otros ya cometieron, también evita comenzar desde cero cada vez.

Hay dos escenarios para el reuso de la experiencia. El primero es cuando hay un canal unidireccional del productor de la experiencia al consumidor (usuario) de la experiencia y ambos están claramente separados; en este escenario los productores de la experiencia en sí podrían ser recompensados por su trabajo, y por lo tanto la experiencia convertirse en una mercancía comercializable. En el segundo escenario, la experiencia se comparte para el beneficio de un grupo de personas, por ejemplo, los trabajadores de una compañía o un grupo de usuarios. En este arreglo, no hay una diferenciación explícita entre los productores y consumidores de la experiencia, y la experiencia se comparte en algún tipo de almacén central. [Ber02]

5.1. Beneficios del reuso de experiencia

Reusar la experiencia puede dar algunos de los beneficios:

Acortar los ciclos de resolución de problemas El reuso permite reducir el costo de la resolución de problemas, el tiempo de solución.

Mejorar la calidad de la solución Una solución de mejor calidad se puede cotizar mejor en el mercado, y puede mejorar el prestigio del solucionador de problemas.

Disminuir el nivel de habilidades necesarias para resolver un problema Dado que el solucionador de problemas puede acceder a la experiencia de otros, esta experiencia es una forma de ayuda que le permite resolver problemas que no podrían resolver solo y también se reduce el costo y tiempo de entrenamiento de los usuarios.

5.2. Procesos de la administración de la experiencia

La administración de la experiencia una forma limitada de la administración del conocimiento porque está restringida al conocimiento adquirido situado en un contexto particular. La administración de la experiencia se centra específicamente en recolectar, modelar, almacenar, reusar, evaluar y dar mantenimiento a la experiencia.

Recolectar la experiencia Para que la experiencia pueda ser compartida primero debe ser recolectada. Algunas formas de experiencia pueden ya estar disponibles como documentos o dentro de bases de datos, sin embargo otros tipos de experiencia sólo están en la memoria de los expertos, y deberá ser recolectada primero para poderse reusar. Se necesitan mecanismos especialmente para capturar la experiencia conforme esté disponible.

5. La administración y el reuso de la experiencia

Modelar la experiencia Para administrar la experiencia hay que modelarla. El modelado es la actividad clave para seleccionar experiencia reusable y poderla reusar e implica encontrar formas apropiadas de representarla y formalizarla, si es necesario. Diferentes tipos de experiencia y de problemas pueden requerir diversas formas de modelado.

Almacenar la experiencia La experiencia recolectada se deberá almacenar para su uso futuro. El medio de almacenaje puede ser centralizado o distribuido.

Reusar la experiencia El reuso de la experiencia es el objetivo principal de la administración de la experiencia y requiere:

1. Tener acceso a la experiencia y seleccionar la experiencia adecuada para ser reusada.
2. Evaluar la utilidad de la experiencia seleccionada en el contexto del nuevo problema.
3. Si es necesario, adaptar la experiencia al nuevo contexto.
4. Finalmente, apoyar al usuario para resolver el nuevo problema al reusar la experiencia.

Evaluar la experiencia Durante el reuso de la experiencia ésta puede ser evaluada en el contexto del nuevo problema a ser resuelto. La evaluación puede ser en términos de lo apropiado que fué la experiencia para el problema que se está resolviendo, o en términos de la exactitud y actualidad de la experiencia encontrada. Esta evaluación es importante para mejorar continuamente el proceso del reuso de experiencia.

Dar mantenimiento a la experiencia El último paso es mantener constantemente actualizada la experiencia. Debido a los rápidos cambios en el entorno, la experiencia puede tener una vida útil limitada, la experiencia obsoleta deberá de ser identificada para ser actualizada o removida. También podría ser actualizada la forma que se modela la experiencia. Este proceso se puede iniciar por una mala evaluación de la experiencia o de una forma preventiva.

En este capítulo hablamos acerca de la administración y el reuso de la experiencia. Ahora daremos un giro para contemplar algunos de los aspectos de la motivación de las personas que tienen relación con las mediciones.

6. Relaciones de la medición con el funcionamiento de los equipos de desarrollo

6.1. Aplicación de la medición en el trabajo de los equipos

La aplicación de la medición de software no es una disciplina aislada. Su valor se maximiza cuando se considera en el contexto de otras fuentes de información. Aún más, integrar la medición en otros sistemas organizacionales ayuda a perpetuar el programa de medición por un largo periodo.

Las mediciones a nivel de proyecto son el cimiento para las mediciones a nivel de proceso y de organización. La mayoría de los datos se origina en un nivel de proyecto. Establecer un buen programa de medición permite disponer de datos necesarios para evaluar el desempeño de los procesos y de una organización multi-proyectos. Usar datos de proyectos para ayudar a la toma de decisiones de procesos y organizacionales incentiva la recolección y análisis a nivel de proyecto. Esto se tiene que tomar siempre a la luz de las salvedades enunciadas en el Apéndice B.

Para comprender cabalmente los siguientes conceptos, habrá que hacer un pequeño repaso por algunos aspectos de la filosofía empresarial.

6.1.1. Tipos de organización

Existen dos tipos de compañías comerciales, distinguibles por su razón primaria para realizar negocios.

El primer tipo de empresa está operada con un propósito puramente “económico”: producir los máximos resultados con los mínimos recursos [MSK02]. Este tipo de “compañía económica” está gerenciada principalmente para obtener beneficios. La gente es considerada como “activos” – extensiones de los activos de capital de la empresa. Igual que con los activos de capital, la inversión en activos humanos se reduce al mínimo necesario en una compañía económica, para producir el mayor retorno posible en el periodo de tiempo más breve.

La compañía económica no es una comunidad de trabajo, excepto incidentalmente. Es una máquina corporativa. Su único propósito es la producción de riqueza para un pequeño grupo de gerentes e inversores. No siente ninguna responsabilidad por el personal en su conjunto. No brinda una comunidad para sus empleados y gerentes, excepto como un subproducto de su propósito primario: el retorno sobre la inversión en activos de capital y bienes humanos.

Se deben reforzar los controles jerárquicos; de otra manera, la máquina económica no trabajará de manera efectiva. Llevar a cabo rígidos controles jerárquicos significa que se reducirán los márgenes de utilización del talento y la creatividad de todos los que trabajen para la empresa.

En contraste, el segundo tipo de compañía, está organizada sobre un propósito de perpetuarse como una comunidad en marcha. Este tipo de empresa tiene la longevidad de un río. Una compañía mediante la elaboración de reglas y el movimiento de su gente, puede emular la longevidad y pujanza de un río.

En este tipo de “empresa-río”, el retorno de inversión sigue siendo importante. Pero los directivos consideran la optimización del capital como un complemento de la optimización de la gente. La compañía en sí misma es una comunidad primariamente, sus propósitos son la longevidad y el desarrollo de su propio potencial, y la rentabilidad es un medio para ese fin. Y para producir tanto beneficios como longevidad, se debe tener cuidado con los distintos procesos necesarios para construir una comunidad; definir la pertenencia, establecer valores comunes, reclutar la gente, desarrollar sus capacidades,

evaluar su potencial, cumplir con el personal, manejar las relaciones con extraños y contratistas y establecer políticas para dejar la empresa elegantemente.

Al mismo tiempo, sin embargo, mantiene su identidad cohesiva.

6.1.2. Estilos gerenciales

McGregor [McG85], estudió a los gerentes desde el punto de vista de la motivación humana. Entre sus conclusiones están que hay fundamentalmente dos estilos gerenciales, que operan desde dos diferentes conjuntos de supuestos.

El primer tipo de gerente opera con los supuestos que McGregor llama *teoría X*, en la que creen que a la gente:

- Le disgusta el trabajo
- No se puede confiar en ellos y tienen que ser vigilados continuamente
- No pueden tomar buenas decisiones sin supervisión estricta
- Necesitan ayuda para completar hasta las tareas más sencillas

Un gerente que cree en estas cosas naturalmente cae en sospechas, un estilo coercitivo, donde la supervisión estricta y los castigos son la regla. En contraste, los gerentes que creen lo que McGregor llama la *teoría Y*:

- Disfrutan el trabajo
- Son dignos de confianza y trabajan aún sin supervisión
- Pueden tomar buenas decisiones autónomamente
- Pueden terminar sus tareas asignadas, sin ayuda continua

Debido a estas creencias, los gerentes de la teoría Y apoyan una forma de administración enfocada a las metas y las recompensas.

Parece curioso que puedan coexistir éstos dos puntos de vista simultáneamente, si se supone que los gerentes no son tontos y es probable que adapten su estilo con el tiempo para mejorar. ¿Cómo es que coexiste a largo plazo? Una respuesta es que ambas estrategias son localmente óptimas. Una solución lo es cuando una pequeña variación en las variables de decisión de las condiciones actuales de operación resulta en una baja en el desempeño; por lo tanto los gerentes adeptos tanto a la teoría X como a la teoría Y encuentran que casi no hay forma de mejorar su desempeño en la inmediata vecindad de la solución adoptada.

Se podría pensar que se tienen dos soluciones diferentes, cada una con sus propios méritos. Pero como Corregir argumenta, la teoría Y es la mejor de las dos porque apela a los más altos motivos al enfatizar la gratificación del ego. Ésto es significativo, ya que una vez que los empleados toman la responsabilidad de vidas en sus propias manos y buscan autónomamente metas que los enriquecen como individuos y como organización en su conjunto, los gerentes de la teoría Y tienen menos problemas para mantener funcionando bien a la empresa.

En contraste, la situación de los gerentes de la teoría X, en donde para mantener la empresa funcionando tienen que aumentar las amenazas para mantener los mismos (si no es que disminuyen) niveles de desempeño. Los esfuerzos de supervisión funcionan mientras los gerentes de la teoría X están ahí parados, pero en cuanto vuelven la espalda las tropas buscan formas de sabotear las cosas, echarlo a perder o de alguna forma obstruir el progreso. Con el tiempo el estilo coercitivo de la teoría X requiere una escalación de amenazas que son cada vez menos efectivas.

6.1.3. La teoría Z

Mientras que la teoría XY de McGregor está enfocada al estilo gerencial y la motivación desde el punto de vista de la organización y los gerentes, la teoría Z pone más énfasis en la actitud y responsabilidades de los trabajadores. La teoría Z no es una idea de McGregor, ni es una extensión de la teoría XY. Una de sus dos variantes fue propuesta por Abraham Maslow en su libro *Maslow on Management*, la otra variante fué hecha por el Dr. William G. Ouchi, basado los "14 puntos" del Dr. W. Edwards Deming y se nombra normalmente como el estilo de administración "Japonés". [teo]

La teoría Z de Maslow presupone que una vez que una persona ha alcanzado un cierto grado de seguridad económica, desea una vida basada en valores, una vida laboral donde la persona pueda crear y producir.

Uno de los conceptos más importantes de esta teoría, es que la dirección debe tener un alto nivel de confianza en sus trabajadores para que este estilo gerencial funcione. Esta teoría asume que los trabajadores tendrán una gran participación en las decisiones de la compañía, para que ésto funcione los empleados deben de tener mucho conocimiento acerca de los asuntos de la compañía, y poseer las competencias para tomar estas decisiones. Por esto, bajo la teoría Z enfatiza las necesidades de que los trabajadores que se conviertan en generalistas, en vez de especialistas.

Los puntos en común entre la teoría Z de Ouchi y de Maslow no son coincidencia, dado que los tratados de Maslow en administración y motivación fueron fundamentales para Deming y Drucker que reconstruyeron Japón después de la Segunda Guerra Mundial, y ésto éxito al administrar a su personal es el tema del libro de Ouchi [dem,teo]. La semejanza de la teoría de Servitje a las teorías Z es debido a que está basada en lo que se documentó acerca de las empresas japonesas y lo que observó personalmente en Japón.

6.1.4. La participación en la empresa

Que el trabajo no sea sólo una tarea, sino una misión, una pasión y una aventura. Para ello las metas de los trabajadores tienen que ser las mismas de quienes la dirigen. Sólo así asumirán plenamente su responsabilidad y darán lo mejor de sí mismos. Lorenzo Servitje [Ser03]

El sistema de contrato y salario, de división del trabajo, de poca información y de escasa o nula participación no encaja con la naturaleza humana; la contradice y la violenta, por eso no funciona. Es ineficiente, no permite la realización del hombre, sino que propicia su enajenación.

Las empresas japonesas funcionan como grandes familias: todos están interesados no en los beneficios sino en el bien común y el trabajo en equipo. Hay muchas virtudes en este estilo de trabajar que debemos estudiar y adoptar si queremos permanecer y mantener nuestro liderazgo; además, funcionan con un estilo más humano y fraternal, con el que todos nos sentimos satisfechos.

El éxito del empresariado japonés se debe más que nada al respeto a las personas. No menosprecia a su personal cuando quieren hacer algo; lo involucran. Los obreros japoneses y asiáticos en general, se sienten parte de su empresa; se les respeta e informa; se les permite participar. Y ellos se entregan con el alma. Piensan que su empresa es un lugar que tienen que cuidar; que es parte de sí mismos; que no les va a fallar. Lo más importante en esto es que en vez de usar a las personas, las involucra.

La participación no es un objetivo, es una herramienta para dar verdadero contenido y hacer realidad el anhelo de que los trabajadores no sean ejecutores silenciosos sino que se empapen de los objetivos y las necesidades de la empresa, para que con interés, creatividad y capacidad se integren al esfuerzo general.

La participación es la manera de hacer realidad el principio de subsidiaridad indispensable para que la gente crezca y se interese, para que no sea manipulada ni usada.

La participación se presenta en un esquema sencillo con seis pasos que hacen la realización de este proceso se lleve a cabo de forma lógica, armónica y eficaz:

6. Relaciones de la medición con el funcionamiento de los equipos de desarrollo

- Participación en la información
- Participación en la consulta
- Participación funcional
- Participación en las utilidades
- Participación en la propiedad
- Participación institucional

6.1.4.1. Participación en la información

La participación en la información significa que las personas en la empresa –pequeña, mediana o grande- sepan todo lo que necesitan saber. Que conozcan los planes, los resultados, los problemas, los proyectos, incluso los sueldos. Es muy difícil dar toda la información.

Primero porque abrirla cuesta trabajo: se necesitan tableros, boletines, juntas. Se requiere también una estructura organizacional que permita la administración en varias direcciones: de abajo hacia arriba, de arriba hacia abajo y de forma lateral. Se necesita orden. Otro aspecto de suma importancia es que es indispensable una transparencia total para poder hablar. Se necesita poder decir todo a las personas adecuadas de la manera adecuada y en momento adecuado.

En efecto, este primer paso es el más difícil porque presupone una decisión auténtica de avanzar hacia un plan de participación.

Por otra parte, informar, en el sentido que aquí se entiende, implica cuatro pasos que pueden llevarse a cabo y que tampoco son fáciles:

- Encontrar los medios físicos para hacerlo
- Decir siempre la verdad
- Conseguir credibilidad entre los trabajadores
- Informar, transmitir información ordenada, sistemática, que llegue a todos, sin deformaciones.

Los medios para proporcionar esta información depende en gran medida del tamaño de la empresa.

La veracidad y la totalidad de la información son requisitos indispensables. Es necesario comunicar metas, planes, objetivos, resultados, problemas, oportunidades y toda la información que se requiera para evitar rumores y temores. Hay que hacer saber los éxitos y los fracasos, hay que involucrar a las personas en las estrategias para que, consciente o inconscientemente, todos tiendan hacia el objetivo común.

6.1.4.2. Participación en la consulta

La consulta es apenas la consecuencia lógica de una buena implantación del primer paso. Si las personas están bien informadas, si confían en las situaciones expuestas, es evidente que la siguiente medida que conviene tomar es preguntarles su opinión sobre los diversos aspectos. Se abre un canal de comunicación con fundamentos. Asimismo, seguramente la empresa se interesará por conocer la manera en que se perciben las situaciones y por las posibles ideas o soluciones que puedan ofrecer. Lo que en un principio fue información, empieza a convertirse en comunicación.

Una vez que los trabajadores están informados, podrán contar con las herramientas para opinar sobre temas referentes a la empresa. ¿Quién mejor que el mismo trabajador sabe de sus labores? En esta etapa, los trabajadores deben ser consultados acerca de diversos temas. Igualmente, antes de tomar una decisión, los jefes deberán preguntar a sus colaboradores qué opinan sobre las acciones que

la empresa pretende emprender. Si bien ellos no serán quienes decidan, su punto de vista será tomado en cuenta.

Si el trabajador aporta opiniones, ofrece soluciones, participa con sus respuestas, pregunta y al final del proceso ninguna de sus opiniones se usa, puede sentirse mal y dejar de hacerlo.

Si el jefe toma en consideración la opinión de alguien, será importante entrenar a esa persona, con el propósito de que los demás sepan que sí se están escuchando las aportaciones, y para motivarles a que sigan participando de manera dinámica y continua.

Cada etapa sirve como base para la siguiente. Por ejemplo, en el tercer paso, la participación funcional, el trabajador deberá estar seguro de sus opiniones y disponer de toda la información referente a su trabajo y las relaciones del mismo.

6.1.4.3. Participación funcional

Se entiende por participación funcional la involucración y el conocimiento del trabajador de los aspectos de su área laboral que le permitan realizar su trabajo y tomar en lo personal, o en grupo, ciertas decisiones o iniciativas.

La lógica dicta que el trabajador que realiza continuamente una función, sabe de ella más que muchos otros; asimismo, si conoce los objetivos, si está capacitado y motivado, es posible y probable que pueda corregir sus errores y modificarla o tomar las decisiones que mejor convengan cuando sea necesario.

Cuando los trabajadores están capacitados, informados, motivados y cuentan con la libertad de usar su iniciativa de ser necesario, los resultados son muy positivos. Muchas funciones que por estar hoy reservadas a los jefes provocan resentimientos en aquellos de menor jerarquía, pueden ser manejadas por equipos responsables. En cambio, cuando los trabajadores no están fuertemente involucrados y responsabilizados del objetivo de su esfuerzo, se generan resultados negativos, como la ineficiencia, la burocracia y la mediocridad.

Una sana aplicación de dicho principio se convierte en el principal motor para el desarrollo de las personas. Les da confianza en sí mismas; descubren que pueden ser creativas; florecen sus iniciativas; usan sus capacidades latentes, y son más productivas en beneficio de ellas mismas, de su empresa y de su comunidad.

Sin la aplicación de este principio, se mantiene a las personas como menores de edad, sin responsabilidades, entusiasmo ni confianza en sí mismas. Por lo tanto, se obtienen resultados mediocres.

La participación funcional no es una panacea, pero es una de las herramientas que permiten la transformación de las actitudes y resultados del trabajo.

6.1.4.4. El contexto del empowerment

Desde una perspectiva histórica, las innovaciones en el nivel de grupos de trabajo se puede ver como una tendencia con largo tiempo. Antes de la revolución industrial, las mercancías las hacían artesanos que tenían la responsabilidad del proceso completo. Incluso a principios del s.XX los automóviles eran construidos por artesanos calificados que planeaban la producción, resolvían problemas de diseño y construían el auto como una unidad. En los años 20's las ideas de F. W. Taylor, el padre de la administración científica, influenciaron ampliamente a la dirección a dividir los trabajos en tareas pequeñas y decidir el mejor método para realizar cada tarea usando métodos de estudio del trabajo. Bajo éste régimen, los trabajadores tenían poca influencia con la separación de la concepción separada de la ejecución, y el trabajo intelectual se centraba en la dirección. El sistema se basaba en la obediencia de los trabajadores. Aunque la administración científica fue muy exitosa al elevar la productividad, había una preocupación con la alineación de los trabajadores reflejada en alta rotación de personal, ausentismo y conflictos. El trabajo de Elton Mayo y la Escuela de Relaciones Humanas criticó el Taylorismo y sugirió que involucrar a los trabajadores tendría grandes beneficios de negocio y morales. Los trabajadores estarían auto-motivados y harían bien el trabajo con poca supervisión. [Wil97]

6. Relaciones de la medición con el funcionamiento de los equipos de desarrollo

Con la evidencia de los problemas de las formas tradicionales de la organización del trabajo, ha habido un interés continuo en involucrar más a los trabajadores, aunque el tipo de iniciativas ha ido y venido con el tiempo. En los 60's se estableció el "enriquecimiento laboral" como un paradigma alterno del trabajo, teniendo como meta dar a los empleados un trabajo significativo con cierto grado de control y de retroalimentación en la eficiencia. La motivación intrínseca se veía como un factor crítico para la satisfacción laboral y los trabajos se debían de "enriquecer" al reintegrar tareas de mantenimiento y dar algunas oportunidades de toma de decisiones. Walton cita compañías como General Motors, Procter & Gamble y Mars como líderes en la innovación laboral en EE.UU. En los 70's hubo un mayor interés en la democracia industrial con un énfasis en los derechos de participación de los trabajadores, y el apoyo legislativo para directivas del trabajo en Europa Occidental dio un empuje a tal estructuración.

Para los 80's se desarrollaron nuevas formas de participación, menos interesadas con el concepto de negociación conjunta y con mucho más énfasis en involucrar a los empleados como los círculos de calidad, team breafing y compartir las ganancias como parte de una serie de reformas a las prácticas laborales. El punto de estos esquemas, es que no ponen en duda a las prerrogativas de la dirección.

Fue al final de los 80's que el empowerment tomó su forma actual. El rumbo del empowerment encajaba con nociones de cultura empresarial que ven a los individuos como emprendedores tomando su destino en sus propias manos que ya no estaban atadas por reglas burocráticas ni obstrucciones sindicales.

6.1.4.5. Facultamiento (empowerment)

Empowerment (facultamiento), o darle poder al colaborador, significa asignarle la responsabilidad de realizar tareas, pero también otorgarle autoridad. [Ser03]

El dar la oportunidad de tomar decisiones o empowerment es reflejo de la filosofía antes analizada. Los trabajadores responsables de su propio trabajo siempre han estado motivados y son eficientes. No se usa mucho porque la prioridad de la mayoría de los líderes es el control, pensando que sólo sus ideas son la causa de la eficiencia. El ritmo de cambio rápido de tecnología fuerza a la adopción de los trabajadores responsables. El empowerment se implementa cuando la eficiencia tiene prioridad sobre el control.

Se ha observado que a los trabajadores se les califica como "recursos humanos", lo cual inconscientemente trata al hombre como un recurso. Se dialoga con líderes sindicales pero no con los trabajadores, se negocia desde posiciones de fuerza, se imponen castigos y las metas estándares y procesos se fijan desde arriba, dejando al trabajador como poco menos que un ejecutor silencioso.

El verdadero empowerment mueve la responsabilidad hacia debajo de la cadena de mando, hasta llegar a las trincheras (front line). La verdadera educación son los compañeros de equipo compartiendo conocimiento para lograr una meta común. La motivación se relaciona con la participación en la toma de decisiones. La gente motivada encuentra formas de hacer lo imposible y como resultado tiene alta eficiencia.

Para liberar el enorme potencial de una persona, hay que tratarla como persona, dialogar e involucrarla, darle la opción de participar, la oportunidad de innovar, de cometer errores y de aprender de ellos, o sea, la posibilidad para todos de ser y sentirse parte del equipo.

6.1.4.6. Participación en las utilidades

La justicia y la equidad señalan que los frutos de la productividad debe compartirse, y una manera sana es hacerlo entre todos los factores de la producción: con los inversionistas, mejorando el rendimiento de sus fondos invertidos, con los clientes o consumidores, mejorándoles el precio y calidad y con los trabajadores, mejorando sus ingresos.

6.1.4.7. Participación en la propiedad

Hablar de participación en la propiedad es un tema complicado y se asocia con prácticas socialistas. Existen diversas formas de participación en la propiedad. El primer caso son las cooperativas, en las que todos los integrantes de la empresa son participantes de alguna parte del capital. No parece que puedan proliferar, ya que el liderazgo necesario para formar una sociedad no parece abundar.

En algunos países con mercados de valores desarrollados, los trabajadores obtienen acciones de sus propias empresas, ya sea a través de planes de opción de acciones, o comprándolas directamente. En muchas de éstas empresas, este tipo de participación se limita a los funcionarios de más alto nivel.

Hay empresas que nacen como una sociedad en la que todos sus integrantes son socios en mayor o menor proporción, sobre todo cuando es un grupo de profesionales que se reúne para integrar una empresa de servicio. Otras empresas abren la posibilidad a sus trabajadores de todos los niveles de adquirir acciones de la misma.

6.1.4.8. Participación institucional

En la participación institucional, la representación oficial de los trabajadores, o sea, los sindicatos, forma parte del directorio del consejo directivo de la empresa.

6.1.5.

6.2. Establecer y sostener el compromiso

La medición puede significar un cambio significativo en la forma de hacer negocios de una organización. Las preocupaciones relacionadas con este cambio deben ser abordados directamente.

6.2.1. Tener el compromiso de la organización

El objetivo de la tarea de obtener el compromiso de la organización es desarrollar apoyo para la medición a todos los niveles de la organización. Los programas de medición ordenados por la alta dirección raramente tienen éxito sin que el personal técnico y de proyecto lo tome como suyo. El equipo a todos los niveles necesita entender como la medición beneficiará directamente sus proyectos y sus propios flujos de trabajo.

Implementar un proceso de medición requiere un cambio cultural significativo dentro de una organización. Como cualquier otro cambio, la incertidumbre asociada a su introducción podría causar angustia que se puede aliviar al comunicar al equipo un entendimiento completo del proceso de medición y como se usarán los resultados de la medición a todos los niveles de la organización. Algunas personas pueden pensar que la medición se usará impropiamente para evaluar el desempeño individual. La medición es más efectiva en una organización cuya cultura permite al personal hablar de problemas y riesgos, lo cual puede significar un cambio cultural radical.

Para que un programa de métricas sea exitoso debe de ser adoptado por la alta dirección y por el personal técnico. El soporte, los involucrados y el liderazgo de la gerencia son importantes para una implementación exitosa de un proceso de medición. El apoyo (support) va más allá de que la alta dirección diga que “la medición es una buena idea”, sino que involucra establecer políticas de medición, dar recursos adecuados, crear un foco en la medición al comunicar los objetivos y definir los factores más importantes de éxito de la medición. También involucra revisar los datos de medición y analizarlos rutinariamente buscando retroalimentación y tomando acciones derivadas de las recomendaciones de los análisis.

Muchos gerentes y miembros del equipo tienen su primer acercamiento a la medición cuando un “evento” significativo en el software ponen en duda la forma en que se está administrando un proyecto o la organización. Otros lo tienen como el resultado de una política o iniciativa. Pocos individuos

6. Relaciones de la medición con el funcionamiento de los equipos de desarrollo

por primera vez conocen la medición del software como un proceso para una administración efectiva de proyectos que permite alcanzar objetivos personales, de proyecto y organizacionales. En muchos casos el equipo de proyecto y la gerencia ven la medición como “una cosa más que hacer” y como algo que requiere recursos adicionales que ya de por sí están sobrecargados. Para contrarrestar esta perspectiva, se necesitan identificar claramente los beneficios que tendrá la medición en los niveles organizacionales, de proyecto e individual.

Con un liderazgo efectivo de la gerencia, la organización completa usualmente reconocerá la importancia de la medición y activamente ayudará en el proceso. Involucrar al personal técnico al definir mediciones y los procesos de medición ayuda a promover a que el personal lo tome como suyo. El personal deberá de ser invitado a sugerir datos e indicadores específicos que sea relevantes a sus objetivos y necesidades de información. Estas recomendaciones deberían de usarse para desarrollar mejores métodos de recolección de datos.

El abordar la cuestión de “¿qué tiene que ver conmigo?” también ayuda a tener apoyo a todos los niveles. Ayuda a los individuos y a las organizaciones a poner metas propias y provee un mecanismo para medir el progreso hacia esas metas. La alta dirección beneficia al equipo de proyecto al hacer la labor directiva más proactiva y para proveer la base de una planeación más realista.

6.2.2. Definir las responsabilidades en la medición

El número de personas involucradas y la asignación de las tareas de medición varía considerablemente entre organizaciones, dependiendo del tamaño y de lo que abarque el proyecto. El equipo de medición puede consistir de un analista que dedique tiempo parcial a la medición o un equipo de personas. Un grupo de personas puede organizar un comité directivo, el principal responsable debe ser al nivel directivo, representado por la dirección o por el administrador de proyecto. La medición ayuda al alto directivo a determinar el status de los proyectos individuales y tomar decisiones que se aplicarán en toda la organización.

El administrador de proyecto tiene la responsabilidad directa por el desarrollo de un sistema de software. En muchos casos el administrador de proyecto es el principal usuario de los resultados de la medición, también es responsable de identificar y administrar las necesidades de información del proyecto y de comunicárselas a los niveles superiores de la dirección, también debe de usar mediciones para tomar decisiones de proyecto.

Es responsabilidad del administrador de proyecto asegurarse de que las mediciones estén integradas con el proyecto. La integración incluye todas las actividades de soporte que hacen del proceso de medición parte de los demás procesos técnicos y de administración del proyecto.

Mientras que la alta dirección es responsable de integrar y usar mediciones en la organización, al personal técnico del proyecto usualmente se le asignan las tareas del día a día relacionadas a la planeación y la realización de la medición. Generalmente un individuo o un equipo tiene la responsabilidad de planear la medición, recolectar y procesar los datos, analizar los resultados y reportarlos a la alta dirección.

Para coordinar las actividades de medición, las organizaciones frecuentemente establecen un comité directivo de medición con la representación de varios proyectos y de personal que realiza labores de apoyo. Estas personas pueden incluir al administrador de proyecto, programadores, probadores, e incluso, personal de contabilidad. El comité usualmente tiene la responsabilidad de las políticas, procedimientos y definiciones comunes de medición. Algunas organizaciones podrían ser demasiado pequeñas para poder tener un comité de medición, y en esos casos, los administradores de proyecto tienen que mantener sus propios planes y conjuntos de datos.

Dependiendo del tamaño y del alcance del proyecto, el equipo de medición puede consistir en un analista que dedica tiempo parcial a la medición, o un equipo de varias personas. En cualquier caso, la principal responsabilidad del proyecto debe ser asignada a una persona en específico, y esa persona es la que deberá de interactuar con la organización de desarrollo. Sobre todo, el personal responsable de

la medición deberá ser capaz de dar resultados directamente a las personas que toman las decisiones del proyecto.

Otros miembros del equipo de proyecto técnico del proyecto también tienen la responsabilidad dentro de su propio proceso de medición. Cada uno deberá entender como el proceso funciona y que información les puede dar. También deberán colaborar en el proceso de análisis de las mediciones al ayudar a identificar eventos del proyecto que pueden tener un impacto en la interpretación de los resultados de medición.

Los resultados de la medición son más efectivos cuando son usados por el equipo de desarrollo de software de forma constante. Adicionalmente, al ayudar a comunicar problemas y soluciones, el equipo de desarrollo, usa los resultados de la medición para identificar y corregir rápidamente problemas como parte de su trabajo diario.

6.2.3. Proveer recursos

La experiencia indica que los programas de medición requieren entre 1 y 5 % del presupuesto del proyecto [MCJ⁺02]. El costo real de un programa de mediciones se determina por el alcance de los datos recolectados y la habilidad de los planeadores de integrar la recolección de datos y su análisis en las prácticas técnicas y administrativas existentes.

El costo de medición para un proyecto individual puede disminuir al irse consolidando la medición como una actividad organizacional. Es importante ver al proceso de medición como una inversión a largo plazo; en un plazo relativamente corto de tiempo después de que se establezca, la medición debe de convertirse en autosuficiente, ahorrando al menos tanto como cuesta.

En algunas organizaciones, los costos de medición de los proyectos individuales se podrían reducir al establecer un equipo de medición como un recurso organizacional.

Los costos de la medición incluyen salarios, capacitación y herramientas para generar, procesar, analizar y reportar datos. Los equipos de proyectos con procesos maduros pueden usar internamente datos derivados de la medición para administrar sus proyectos, por lo tanto, el costo adicional de recolectar y reportar estos datos al nivel organizacional debería de ser muy poco. Si un equipo de proyecto no está usando mediciones, sus procesos puede que no sean muy maduros.

6.2.4. Capacitación en medición

El personal a todos los niveles de la organización requiere capacitación que directamente se enfoque en las competencias requeridas para sus roles.

La dirección y los administradores de proyecto podrían requerir entrenamiento en las relaciones entre el proceso de medición y los procesos de ingeniería y los productos de la organización. Éstos gerentes deben entender las fortalezas y limitaciones del proceso de medición para proveer información que se pueda usar para la toma de decisiones.

Los administradores de proyecto deben entender los resultados de medición que les presenten, también entender como interpretar los resultados y tomar acciones efectivas. Así mismo deben de tener confianza en el proceso de recolección de datos.

Los analistas necesitan capacitación o experiencia en habilidades básicas de medición. Tal vez se necesite entrenamiento adicional para tareas más avanzadas de análisis, como estimación, modelado y análisis estadístico.

Los miembros del equipo de proyecto deben de capacitarse como proveedores de datos. No pueden realizar sus roles en el proceso si no entienden los requerimientos de recolección de datos y la forma de reportarlos.

6.2.5. Herramientas de medición

Una vez que se han definido los requerimientos específicos y los procedimientos de medición, se deben identificar las herramientas para recolectar, procesar y analizar los datos. En proyectos pequeños, el proceso de medición se puede apoyar en una PC con software comercial. En proyectos más grandes o aquellos que requieran herramientas de análisis más avanzadas, posiblemente se requieran herramientas más avanzadas. No se debería de comprar una herramienta específica antes de determinar si realmente soportan las necesidades de información de un proyecto. Las necesidades de información y el proceso de medición impulsan los requerimientos de la herramienta, no al revés. Un conjunto existente de herramientas de medición no deberían de usarse para definir un proceso de medición.

Algunos lineamientos generales para seleccionar una herramienta de medición:

- Seleccionar herramientas que se puedan adaptar para cumplir necesidades específicas del proyecto
- Evaluar herramientas que ya se estén usando dentro de la organización
- Al seleccionar herramientas de ingeniería, considerar si habilidad para producir datos técnicos y de administración acerca de los procesos y productos que soportan.
- No se planea un programa de mediciones basado sólo en las herramientas existentes
- Seleccione herramientas que automaticen la mecánica del manejo de datos como la recolección, procesamiento, análisis, importación/exportación y reporte de información.
- Seleccione herramientas que corran en una plataforma común.

Las primeras tres tareas en la actividad de establecer y sostener el compromiso pueden ser conducidas en paralelo en un proyecto. Una vez el personal y los recursos indispensables se han asignado a un proyecto, se puede comenzar la actividad de planeación, aunque todavía no se haya entrenado a parte del personal, o no estén disponibles algunos recursos. Una vez que se inicie la actividad, los resultados se necesitan revisar para asegurarse que la información proporcionada a las personas que toman las decisiones es la que necesitan para tomar las mejores decisiones posibles, de acuerdo a la cuarta tarea.

Un proyecto puede también adoptar un enfoque incremental al implementar inicialmente conforme se trabaja un pequeño conjunto de necesidades de información y mediciones, y luego ir extendiendo el alcance del programa de medición conforme aumenta la capacidad de la organización. La recolección de datos y el análisis debe de comenzarse tempranamente a pequeña escala para demostrar el valor del proceso de mediciones. Incluso implementar un conjunto pequeño de mediciones clave enfocándose en las necesidades más importantes de información producirá nueva y valiosa información. Las revisiones en este entorno deberían de tenerse justo antes de la implementación de las nuevas medidas. Esto es para asegurarse que las medidas existentes todavía sean aplicables y que las nuevas mediciones y procesos no estén en conflicto con las mediciones existentes y sus procesos.

Durante la implementación inicial, es de primordial importancia enfocarse en obtener datos de buena calidad y asegurarse que los resultados de medición no se usen para evaluar personas o comparar proyectos. Los resultados se deberían discutir con los orígenes de datos antes de reportarlos a la alta dirección. La discusión debería enfocarse en mejorar la visión de los procesos y productos y debería relacionar los resultados a las actividades del proyecto. Los resultados de la medición y sus revisiones, deberían de compartirse con todos los involucrados incluyendo la alta dirección, administradores de proyecto y el personal de ingeniería.

6.3. Ventajas de la adopción de la medición

El desarrollo y mantenimiento de software depende de las personas y el desempeño que tengan depende de la motivación de cada uno. Debido a un vínculo más estrecho con la motivación, la

medición probablemente tenga un mayor potencial para alterar el desempeño de una organización productora de software que otros caminos más comunes, como nuevas herramientas.

Un programa de medición exitoso, es más valioso que el proceso eficiente de datos o los cálculos complicados. La medición es una herramienta para cumplir metas, y no debe ser confundido con “lograr los números”.

La medición de software sólo aproxima los objetivos deseados. Un objetivo difícilmente podría medirse directamente. Incluso un objetivo como “mantener el costo conforme a presupuesto” puede ser engañoso. Aún cuando el proyecto podría estar cumpliendo su meta de costo, el negocio podría fallar porque no se cumplió el verdadero objetivo de capturar un segmento particular de mercado. Todavía peor, la verdadera situación podría no reconocerla la organización si el costo transferido y la reducción de funcionalidad no son decisiones visibles. La cultura de la administración cuantitativa usa la medición para comprender la verdadera situación de forma que se puedan tomar buenas decisiones de forma consciente, y no estar creando objetivos que incentiven acuerdos encubiertos que provocan una sub-optimización del desempeño.

La cultura de la administración cuantitativa de hecho implica un nuevo rol para el administrador de proyecto. En vez de sólo estar monitoreando y reaccionar a las anomalías, tiene que usar las mediciones para investigar y comprender el desempeño del proyecto para que se puedan tomar acciones correctivas oportunas. El administrador de proyectos debe coordinar estas acciones con la alta dirección al comunicar proactivamente el status y los problemas.

6.4. Criterios de éxito de un programa de medición

Establecer una cultura para el uso efectivo de la medición no es fácil. No hay estrategias simples. Sin embargo, se puede decir que se está teniendo éxito si:

- La recolección de datos es natural y automática
- Los datos son fácilmente accesibles
- La gente usa los datos como base para sus decisiones
- Las fallas llevan a un mejor entendimiento, más que a echar culpas
- Los objetivos numéricos están acompañados de planes racionales
- Se hacen mejoras periódicamente al plan de medición

Esto también sugiere un proceso de experimentación. Ningún análisis ni discusión podrá identificar la medición perfecta a priori, incluso la “mejor medición disponible” puede cambiar conforme el proyecto pasa por su ciclo de vida y se gana más conocimiento. Las mediciones deberán evolucionar conforme las necesidades evolucionan. Tal vez las palabras “flexibilidad progresiva” sean las que mejor describen un programa de mediciones – el uso real es lo que impulsa los cambios en las técnicas y análisis de medición. Un programa de medición estático terminará extinguiéndose por mal uso o falta de uso. Lo cual sugiere que es poco probable definir el programa perfecto de mediciones desde el principio del ciclo de vida de un proyecto. Es mejor seleccionar unas pocas mediciones útiles y empezar a realizarlo, que estar eternamente eligiendo entre las opciones de medidas. Una vez andando, el plan de medición evolucionará con la experiencia para cumplir mejor con las necesidades de los usuarios.

Tras haber hablado de algunos aspectos de la conducta humana que influyen en las mediciones, en el siguiente capítulo haremos una síntesis de todo lo expuesto, de como se relaciona con la herramienta propuesta, los beneficios planteados de este enfoque y la arquitectura de la herramienta.

7. Objetivos de la herramienta

7.1. Antecedentes

Las personas que están encargadas de llevar la administración de los programas de métricas han tenido que valerse de las herramientas que conocen y que tienen a la mano, dado que normalmente tienen una carga de trabajo bastante fuerte.

La falta de recursos comprende a recursos tanto humanos como tecnológicos y a factores de la cultura organizacional.

7.1.1. Aspectos técnicos

Dados los pocos recursos que un proyecto de medición enfrenta, tradicionalmente las herramientas más usadas para la recolección y análisis de datos cuantitativos han sido las hojas de cálculo. Sus principales ventajas han sido el conocimiento que los usuarios tienen de su operación, la flexibilidad que le proveen a los usuarios, así como su ubicuidad, ya que en casi cualquier computadora portátil o de escritorio tiene instalado algún paquete de productividad de oficina que incluye una hoja de cálculo.

Las personas que están encargadas de llevar la administración de los programas de métricas ha tenido que valerse de las herramientas que conocen y que tienen a la mano, dado que normalmente tienen una carga de trabajo bastante fuerte. Esto dificulta que puedan construir herramientas más especializadas que podrían beneficiarlos, además que no necesariamente tienen la destreza técnica para construirla, ni los recursos económicos para contratar la construcción de una herramienta a su medida.

A su vez, las personas que trabajan en los proyectos normalmente no le ven mucha utilidad a la medición, porque tienen que realizar la tarea de reportar los datos, lo cual les quita tiempo, pero no tienen una percepción de la utilidad de esos datos, porque no tienen acceso a datos más que a los suyos propios.

La falta de herramientas para la recolección, análisis, publicación y conservación de datos de medición, afecta a un amplio segmento de la industria, no sólo del software, sino en sectores tan diversos como la manufactura, servicios, el gobierno o el agro, básicamente a cualquier persona u organización que desee implantar o tenga un programa de calidad.

En muchos de estos sectores la recolección, el intercambio y el análisis de los datos se da por métodos rudimentarios, de forma fragmentada, esporádica y poco organizada, posiblemente sólo por la iniciativa de alguna persona. Es virtualmente imposible para los involucrados en el proceso (o incluso para un tercero externo) tener acceso a todos los datos primitivos, a los análisis que les fueron aplicados o a poder hacer nuevos análisis a partir de los datos ya existentes.

Aunque en el mercado existen excelentes herramientas de análisis y personal capaz que podría sacarles un provecho insospechado, el hecho de sólo poder acceder a datos derivados nulifica su utilidad, ya que existe siempre la duda acerca de los supuestos que se tomaron para el análisis, la validez del muestreo realizado y de si los datos derivados disponibles fueron calculados con apego a la correcta aplicación de métodos matemáticos, estadísticos y/o lógicos¹. Esta es la razón del elevado escepticismo

¹Por supuesto que hay ciertas organizaciones que disfrutan precisamente de este prestigio, como el grupo de Barry Boehm en USC, Standish group, Gartner, y diversas organizaciones que hacen encuestas y reportes, en los ámbitos políticos, económico, de opinión pública o en industrias específicas. Han podido construir este prestigio gracias a

7. Objetivos de la herramienta

prevalente hacia las estadísticas en general.

Tampoco está estructurado el proceso de conservación a largo plazo. En general los esfuerzos empíricos para recolectar información no contemplan la utilidad a largo plazo de la información, ni los retos que su conservación representa. Incluso las mejores organizaciones si no hay una visión de conservación de largo plazo, la información se terminará perdiendo ², incluso si el personal involucrado está consciente de su posible valor en el futuro.

7.1.2. Aspectos humanos

Otro de los problemas que enfrentan los programas de métricas es la carencia de recursos humanos capacitados.

En general, las mediciones de software no están incluidas dentro del currículum de los centros de enseñanza, lo cual provoca que sea una novedad llegar a un sitio que tiene o quiere implementar un programa de medición. Ésto tiene una vertiente positiva y otra negativa. Por el lado positivo, la mayoría de la gente no está predispuesta negativamente hacia las métricas, y puede que con una introducción pueda encontrarlo interesante, incluso intrigante. Por el lado negativo, como nunca lo han hecho ni visto, no saben que tienen que hacer.

7.1.3. Aspectos organizacionales

Por otra parte, otra de las barreras para la diseminación de información, puede ser la cultura de las organizaciones. Es común que exista una cultura de control en donde los tomadores de decisiones reconocen el valor de la información y por ello se opta por tener un estricto control sobre ella, fomentando el flujo de la información hacia arriba, restringiéndolo los flujos laterales y suprimiendo los flujos hacia abajo [GR95]. En una organización con una administración tipo X [McG85], es la principal barrera.

En una organización con una visión de fomento al intercambio de información, en donde sus integrantes estén comprometidos con esto, la situación es muy diferente. En este caso el reto principal posiblemente sea hacer que la información “permée” a toda la organización, que las personas se den cuenta de la utilidad de la información que tienen a su disposición, que tengan herramientas para aprender a sacarle provecho a esa información, que encuentren nuevas formas de explotar la información y que estas nuevas formas también permeen a la organización.

Aún en organizaciones que tienen explícitamente una cultura orientada a la transparencia y eficiencia, seguramente habrá algunas personas que no la tengan y se presente el fenómeno de la dificultad de compartir información por estas mismas personas que sienten amenazado su control e influencia sobre cierta parte de la organización al perder el control sobre su “información propietaria”.

Si bien, metodologías como PSP y TSP animan al uso de mediciones a nivel individual y grupal, no es la norma que se manejen estas metodologías en las empresas, y aunque se usen, hay que capitalizar el hecho que la gente toma ciertas mediciones al término del curso, ya que si las mediciones no se usan, la gente dejará de tomarlas.

También desde el nivel 2 de CMMI se impone el requisito de tomar mediciones, pero sólo con el fin de tener un archivo histórico para cuando la empresa llegue al nivel 4, no hay un énfasis en empezar a usar los resultados cuantitativos de forma inmediata.

la alta calidad de trabajo que han realizado, al prestigio de los que lo realizan, y ha sido validado por expertos independientes y por los sectores que son su enfoque y sus clientes. Desgraciadamente dentro de la mayoría de las organizaciones no hay los recursos, ni el personal para llevar a cabo un trabajo de este nivel.

²Basta citar el sonado caso de la NASA que no ha podido encontrar las grabaciones de datos y video del Apolo 11, y básicamente ya las dio por perdidas. Aunque desde el momento en que se originaron esos datos no hubo duda alguna de la significancia histórica que el primer alunizaje tripulado significaba, dentro de la visión nunca estuvo conservar para la posteridad las evidencias del evento, y eso provocó que se perdieran las cintas originales en el maremagnum de los archivos de la institución. [Leb07]

En ambientes geográficamente dispersos (como una fábrica de software) o en una organización muy grande una buena implementación de medición cobra más relevancia, ya que es más difícil “darse una idea de cómo están las cosas” estando poco inmerso en las circunstancias de trabajo de un sitio lejano o incluso de un proveedor externo.

Ésto también provoca que el programa sea más difícil de implementar, dado que los potenciales usuarios están dispersos, y no se conocen entre sí.

7.2. La transformación que ha traído Internet

El Internet es el sistema de información más poderoso jamás inventado capaz de dar a cualquier persona acceso fácil y rápido a una cantidad de información sin precedente.

Consideremos sus predecesores. La afamada Biblioteca de Alejandría (en Egipto) fue construida circa 323 A.C. para un público educado, que de hecho eran muy pocas personas dado que la habilidad de leer estaba deliberadamente fuera del alcance de la población. Durante muchos siglos, los monjes fueron los guardianes del conocimiento escrito, transcribiendo e indexando libros como una forma de interpretar la palabra de Dios. Eran premiados tanto por su habilidad para escribir pequeño, lo cual ahorra papel caro, como por su Fe.

Las primeras universidades abrieron alrededor del siglo 4 D.C., la primera enciclopedia formal no apareció sino hasta el siglo XVI, las primeras bibliotecas realmente públicas aparecieron en el s. XIX y proliferaron en el s. XX. Pero sorpresivamente aparece el Internet que permite consultar desde el pueblo más alejado del planeta, tanta información como puede ser resguardada en miles de bibliotecas. El acceso a la información esta cada vez más cerca de su meta, de exclusivo de unos privilegiados a ubicuo.

7.2.1. Todos pueden publicar, y todos lo harán

Una cosa que se puede aprender el Internet es que la gente tenemos mucho que decir. Usamos el Internet para publicar cosas a un paso asombroso. Se crean 120000 blogs diarios, muchos para una audiencia individual. Más de la mitad son creados por personas de menos de 19 años. En Estados Unidos, casi el 40 % de los usuarios de Internet suben videos, y en todo el mundo cada minuto se suben a YouTube más de 15 horas de video. Internet también es altamente social: alrededor de un sexto del tiempo que la gente utiliza para navegar en Internet está en una red social de algún tipo.

Antes publicar estaba restringido por limitaciones físicas. Tenías que tener una prensa y una red de distribución o un transmisor para poder llegar a cualquier número decente de personas, así que la difusión era la norma. Ahora ya no, hoy la mayoría de las publicaciones las hacen usuarios para usuarios, uno para uno o uno para muchos.

También ya pasó la era en donde la información era más valiosa si la guardaban. En los primeros días, cada documento en el web decía al final “Copyright 1997. No redistribuir”. Ahora esos mismos documentos dicen al final “Copyright 2009. Haz click para mandárselo a tus amigos”. Compartir, no guardar la información, se ha convertido en la práctica usual en la web, de forma que no sólo todos pueden publicar, sino que casi todos lo hacen. Ésto es bueno y malo ya que como todos sabemos Internet está lleno de cosas inservibles. La cantidad de basura es una amenaza que opaca los trabajos de calidad.

Hay millones de personas en el mundo que son auténticamente expertos en su área — científicos, académicos, artistas, ingenieros, arquitectos — pero la mayoría de ellos están demasiado ocupados siendo expertos en sus campos, para también pedirles que sean expertos en el nuestro. Tienen mucho que decir, pero poco tiempo para decirlo.

Los sistemas que facilitan la creación de contenido de alta calidad son indispensables para que Internet siga creciendo, porque sin ellos corremos el riesgo de caer en un pozo sin fondo. Necesitamos

7. Objetivos de la herramienta

hacer más fácil publicar su trabajo para los expertos en los que confiamos bajo un modelo de atribución que sea autenticado y extensible, e incluso les permita monetarizar su trabajo de una forma coherente.

7.2.2. Cuando hay datos abundantes, se privilegia la inteligencia

Dar el poder de publicar y consumir contenido a más personas en más lugares permite a todos iniciar conversaciones con hechos. Con hechos las negociaciones pasan de quién habla más fuerte, para ser quién tiene los datos mejor fundamentados. También puede ser un ecualizador que permite tomar mejores decisiones.

Todos deberían de poder defender sus argumentos con datos, y por ello se propone una herramienta que facilite la toma metódica de mediciones, compartirlas, analizarlas, y diseminar el conocimiento y los beneficios de las métricas.

Uno de los resultados colaterales de la era del Internet es que ha acercado el poder de las herramientas analíticas más poderosas hasta al negocio más pequeño. Tradicionalmente los paquetes de software de negocios trataron a los reportes de datos como componentes de segunda.

Cuando todos los negocios dispongan de forma gratuita y ubicua de datos, la habilidad para comprenderlo y extraerles valor se convierte en la pieza que falta. Y ésto le abre el camino a la inteligencia, y los negocios inteligentes son exitosos, independientemente de su tamaño.

Aún esto subestima el verdadero impacto que tendrá la transición de las aplicaciones y datos a Internet. Como Hal Varian ha apuntado estamos en un periodo de “innovación combinatoria” [Ros09], en donde hay una gran disponibilidad de las partes de diferentes componentes que los innovadores pueden combinar o recombinar para crear nuevos inventos. En el s. XIX fueron las partes intercambiables. En los años 20’s fueron la electrónica. En los 70’s los circuitos integrados. Ahora los componentes de la innovación se encuentran en Internet, mediante la cloud computing [clo], con APIs, software de código abierto y servicios de bajo costo, cobrados en base en uso como los servidores de aplicación de Google y Amazon y otros proveedores de aplicaciones de cloud computing. Hay una abundancia de componentes disponibles para cualquiera se este online.

Todavía tenemos un largo camino para hacer las aplicaciones web suficientemente robustas como para los negocios. Aspectos como la latencia, seguridad y confiabilidad de datos tienen que ser iguales o mejores que las alternativas actuales. La experiencia del usuario necesita ser rápida, fácil y rica. Los usuarios esperan que estas aplicaciones funcionen perfectamente para ellos todo el tiempo, y necesitamos cumplir esta expectativa.

7.3. Objetivos de la herramienta

7.3.1. Propiciar la curiosidad y la búsqueda de alternativas

Las soluciones verdaderamente innovadoras son producto de la curiosidad y la curiosidad se nutre de la diversidad intelectual. Se ha descubierto que las personas curiosas hacen preguntas penetrantes que llegan al meollo del asunto y no se limitan a solicitar datos, sino a ahondar para descubrir las posibilidades, los resultados y las opciones. Quienes sienten curiosidad generalmente aciertan, sin importar el asunto, y esto les da una ventaja competitiva tangible.

Si no se tiene cuidado, los únicos pensamientos que terminan mezclándose e intercambiándose, especialmente en los grupos, son aquellos que ya están en la cultura de la empresa. La mentalidad de la empresa pasa a ser del género de: “Aquí no hay necesidad de hacer preguntas”. La apatía intelectual engendra clones: todas las bocas con el tiempo hablan de lo mismo y todos los cerebros se lavan para poder compartir las mismas ideas.

Es una carga demasiado pesada para el jefe ser el proveedor permanente de ideas. Un equipo de alto rendimiento tiene que aceptar retroalimentación de todos sus miembros, e incluso de personas que no son sus miembros.

Uno de los factores por los que la apatía de la gente tiende a aumentar con el tiempo, es que las sugerencias que da no son buenas sugerencias por falta de información de fondo o de criterio. Dada una cantidad suficiente de episodios en donde las sugerencias son desechadas, el interés de la persona en dar sugerencias irá disminuyendo.

Por eso, para fomentar la generación de buenas sugerencias, estas deberán ser basadas en el análisis de información real, usando un criterio propio.

7.3.2. Conservar y compartir conocimiento

Como ya se mencionó, una de las carencias que tienen en general los programas de medición es la carencia de fuentes de personal experimentado en mediciones, por lo que los recursos humanos se tendrán que formar internamente en la empresa.

También es clara la incomunicación existente entre los equipos de medición entre empresas e incluso dentro de la misma empresa. Dado el uso que se le podría dar a la información si la competencia tuviera acceso a ella, es celosamente guardada para evitar su diseminación fuera de la organización. Esto impide que se forme una comunidad abierta de especialistas en medición de software.

Como un paliativo a esto, sobre todo en organizaciones medianas y grandes, se puede formar una comunidad interna de personal interesado en las mediciones, usando la herramienta como uno de los bloques fundamentales para la infraestructura de retención y comunicación del conocimiento, en este caso específicamente el cuantitativo. Y tendría que ser complementada con otros tipos de repositorios para conocimiento cualitativo.

Una herramienta adecuadamente diseñada, permitiría a los interesados compartir y validar sus hallazgos, sus resultados, sus métodos con otras personas de la misma comunidad. Estas interacciones darían lugar a un desarrollo más acelerado del conocimiento dentro de la comunidad, y también tendría la ventaja de poder educar nuevos miembros, que paulatinamente vayan adquiriendo la experiencia que la comunidad ostenta.

7.3.3. Motivación de los miembros de los equipos al participar de las decisiones

La motivación de un equipo de proyecto aumenta cuando las personas tienen injerencia en las decisiones del proyecto, porque pasa de ser “el proyecto” a “mi proyecto”. Este sentido de pertenencia se puede incentivar al propiciar un modelo directivo más participativo y orgánico, en donde la gente directamente involucrada “tome las riendas de su propio destino”.

La disponibilidad de una herramienta que permita tener acceso a los datos primitivos, así como analizarlos puede ser una valiosa estrategia para reforzar la toma de decisiones autónomas “en el nivel más bajo posible” para cada decisión. También puede usarse como una valiosa herramienta para educar el criterio de los involucrados e irles soltando las riendas paulatinamente.

7.3.4. Balancear el control jerárquico vs. la independencia de los equipos

Sin perjuicio de lo anterior la posibilidad de la disponibilidad de datos tiene diferentes usos a diferentes niveles en la organización.

Uno de los usos que le puede dar en cualquier nivel es como una fuente más de información para monitorear diversos eventos que suceden en los proyectos. Esta forma de utilizar el sistema, permite una forma de vigilancia poco intrusiva por parte de la gerencia media y la alta dirección que podrá intervenir en cualquier momento si detecta una desviación desfavorable o inesperada en los datos.

7.3.5. Integración con otras herramientas de recolección

La herramienta deberá de estar preparada para importar los datos directamente de otros sistemas que tengan alguna capacidad de recolección de métricas, por ejemplo, sistemas de seguimiento de

7. Objetivos de la herramienta

defectos, sistemas de control de cambios, sistemas de control de proyectos y programas, contadores de líneas de código, etc. Para agilizar y automatizar la recolección de los datos.

7.3.6. Integración con otras herramientas de análisis

La herramienta deberá de estar preparada para exportar parcial o totalmente sus datos a otras herramientas, por ejemplo, de análisis, en un formato intermedio o directamente en el formato nativo de éstas a fin que los usuarios puedan usarla de la manera más transparentemente posible, aprovechando la familiaridad y experiencia de los usuarios en la operación de éstas. Además de reducir la curva de aprendizaje, también reduce el costo y potencialmente puede aumentar la capacidad y utilidad de la herramienta .

7.3.7. Integración con otros repositorios de conocimiento

Como ya se mencionó, la herramienta será más útil si es parte de una estrategia explícita para ayudar a una organización basada en el conocimiento.

Ésto significa, entre otras cosas, que el conocimiento debe de poderse almacenar de formas fácilmente accesibles. Por eso la herramienta deberá poderse integrar como un bloque en la estructura de repositorios para la retención de ese conocimiento.

El conocimiento al que esta herramienta está enfocado es cuantitativo (los datos capturados, que son numéricos), y un poco cualitativo (los análisis que algún usuario realizó y guarda dentro del sistema). En general las herramientas más usadas actualmente para conservar el conocimiento en las organizaciones están muy enfocadas al conocimiento cualitativo textual o mixto, por ejemplo: manejadores de contenido (repositorios de archivos), FAQs (listas de preguntas y respuestas), intranets, wikis, etc. Si bien se pueden guardar las hojas de cálculo en un repositorio centralizado de archivos, queda para el usuario la labor de la extracción, uniformización y explotación de los datos, que es lenta y engorrosa. Por ello la necesidad de una herramienta que sea un repositorio de datos cuantitativos.

7.3.8. Fácil acceso independientemente del dispositivo

Dado el auge de internet como medio de disseminación de información a través de dispositivos cada vez más diversos uno de los objetivos que se buscan es evitar un acoplamiento excesivo entre el sistema y el tipo de dispositivo de acceso. Hace pocos años el único medio de acceso hubiera sido una PC de escritorio, pero en esta época, no se puede limitar un sistema por la capacidad de procesamiento del dispositivo (PC, laptops, netbook, smartphone, etc), el tipo de medios de entrada (teclado, reconocimiento de escritura, punteros), o las capacidades de los medios de salida (tamaño o resolución de pantalla, idioma). El sistema deberá ser fácilmente adaptable para acomodar todos estos dispositivos.

Como se dice en [MCJ⁺02]: “Los sistemas web son otra forma cada vez más popular para difundir los resultados de las mediciones”, también se busca romper con las barreras geográficas e incluso temporales al quitar la atadura de la aplicación contra un dispositivo de acceso, convirtiéndola en una aplicación in-the-cloud que viva en internet y tenga acceso inmediato desde cualquier momento en cualquier lugar. [?]

7.4. Desarrollo de la herramienta

7.4.1. Descripción de la herramienta

La herramienta presentada, tiene por objetivo ayudar en la recolección ordenada, sistemática y periódica de datos de mediciones, así como proveer integración con otras herramientas de análisis y conservación del conocimiento de la organización.

7.4.2. Caso de uso general del sistema

Para el modelado de los requerimientos de la herramienta, se decidió por el uso de varios tipos de diagramas UML (Unified Modeling Language). Para el caso de la obtención de requerimientos UML tiene los diagramas de Casos de Uso, que es una forma de mostrar las funcionalidades de las que dispondrán los usuarios (actores) del sistema.

El primer diagrama de casos de uso es el diagrama general de casos de uso, y representa una visión a alto nivel de las funcionalidades que el sistema proveerá, por cada tipo de usuario. Estas funcionalidades después se pormenorizan en subsiguientes diagramas

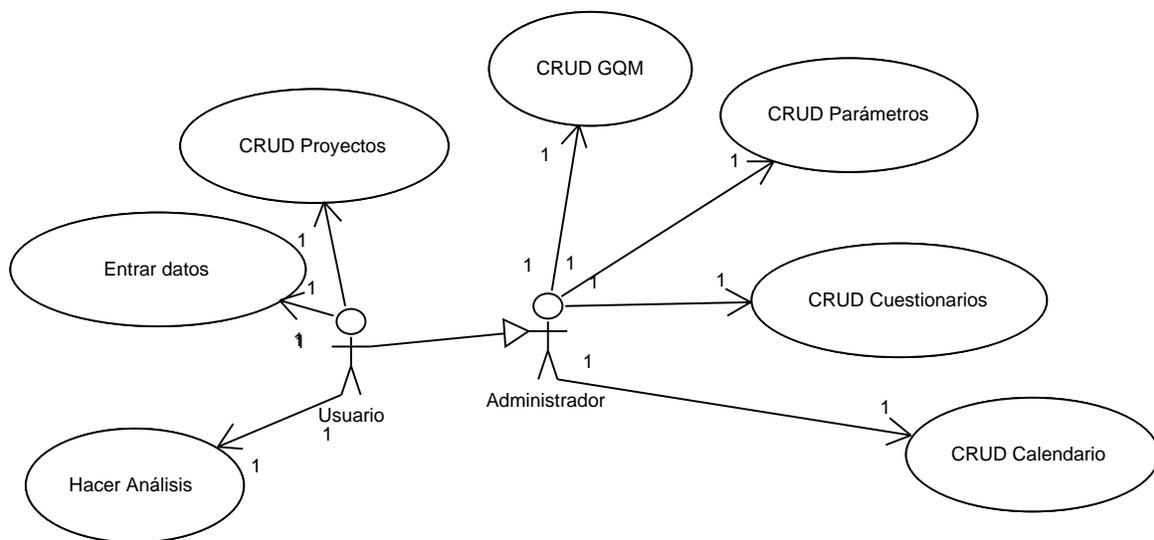


Figura 7.1.: Diagrama general de Casos de uso

7.4.3. Ejemplo de detalle algunos casos de uso

A continuación se ilustra un caso de uso que se eligió como ejemplo para ilustrar la construcción del sistema. El resto de los casos de uso que ilustran la composición del sistema se encuentran en el apéndice A.

La abreviatura CRUD usada en muchos de los diagramas de esta tesis, es el acrónimo en inglés para “Create, Retrieve, Update, Delete”, y es una forma de agrupar los casos de uso que tienen que ver con funcionalidad de edición básica de datos.

7.4.3.1. CRUD Proyectos

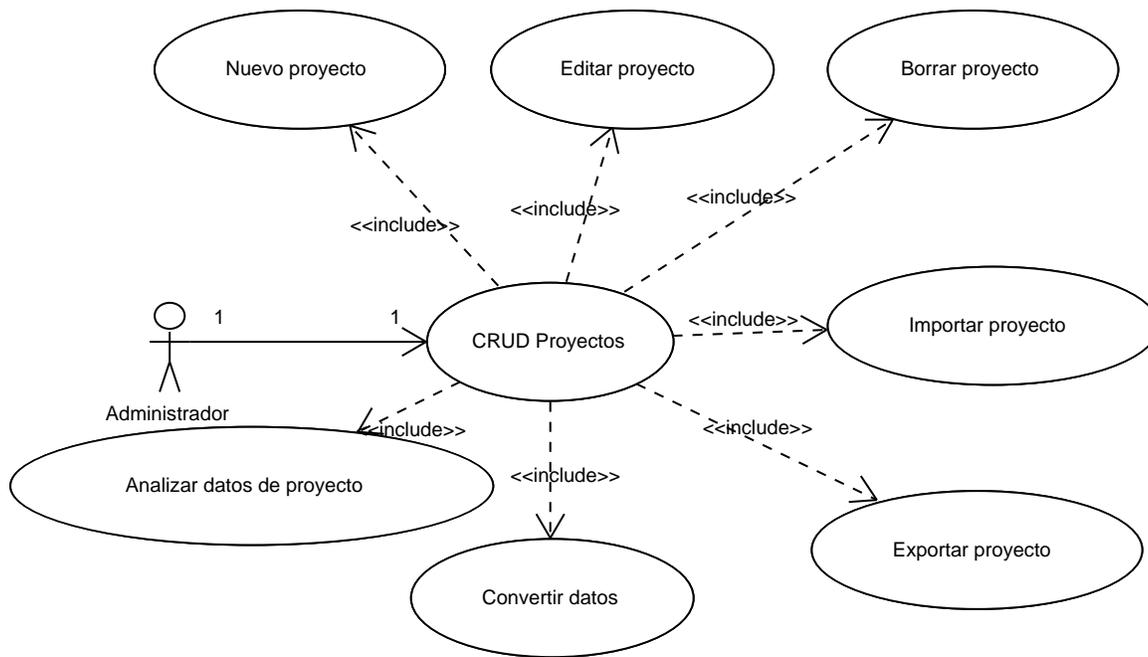


Figura 7.2.: Caso de uso de proyectos

Diagrama

Actor primario Usuario, Administrador

Actor secundario Ninguno

Descripción El usuario desea hacer alguna operación con información relacionada con los proyectos.

Precondiciones El administrador se encuentra autenticado como tal.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción "Proyectos".
2. El sistema le muestra al usuario un menú con los proyectos que tiene registrados el sistema, y un conjunto de acciones que puede realizar con éstos.
3. Si desea agregar un proyecto o importar un proyecto, aquí se puede detener el caso de uso y saltar al subsiguiente.
4. El usuario selecciona un proyecto.
5. El sistema le despliega una panel con el detalle del proyecto al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

7.4.3.2. Seleccionar proyecto

Diagrama Véase diagrama 7.2 en la página anterior

Actor primario Usuario

Actor secundario Ninguno

Descripción El usuario desea seleccionar un proyecto para poder manipular los datos del proyecto con otra funcionalidad del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un proyecto y elige la opción “Hacer activo”. Si el usuario no selecciona un proyecto, el sistema le avisa que debe seleccionar una.
2. El sistema pone como activo el proyecto para esa sesión de usuario y permite que las interacciones con los datos de ese proyecto.

Post-condición El sistema queda con un proyecto seleccionado, al cual se le aplicarán las acciones subsiguientes en el sistema. El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

7.4.3.3. Añadir proyecto

Diagrama Véase diagrama 7.2 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea agregar un proyecto nuevo al sistema.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir proyecto”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda el nuevo proyecto, actualiza el listado de proyecto y pone como activo el nuevo proyecto. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién creado queda como proyecto activo en el sistema.

7. Objetivos de la herramienta

7.4.3.4. Convertir datos

Diagrama Véase diagrama 7.2 en la página 56

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea graduar los datos de un proyecto del sistema al data warehouse.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Convertir datos”.
2. El sistema borra los datos previamente guardados en el datawarehouse del proyecto seleccionado.
3. El sistema guarda los datos del proyecto seleccionado al data warehouse.
4. El sistema muestra una pantalla de confirmación al terminar de convertir los datos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién convertido queda como proyecto activo en el sistema.

7.4.3.5. Exportar datos

Diagrama Véase diagrama 7.2 en la página 56

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea exportar los datos de un proyecto del sistema a un archivo.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Exportar”.
2. El sistema exporta los datos del proyecto seleccionado a un archivo XML.
3. El sistema envía al usuario el archivo XML y muestra una ventana para que el usuario elija el lugar donde guardará el archivo en su disco duro.
4. El usuario elige la ruta y el nombre de archivo.
5. Al terminar de exportar, el sistema muestra una pantalla de confirmación de que terminó de exportar.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién exportado queda como proyecto activo en el sistema.

7.4.3.6. Importar datos

Diagrama Véase diagrama 7.2 en la página 56

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea importar los datos de un proyecto de un archivo al sistema.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Importar”.
2. El sistema le muestra al usuario un formulario en donde puede seleccionar el archivo XML que desea que el sistema importe.
3. El usuario elige la ruta y el nombre del archivo XML que desea importar y oprime continuar.
4. El sistema recibe del usuario el archivo XML y realiza la importación y sincronización de los datos.
5. Al terminar, el sistema muestra una pantalla de confirmación de que terminó de importar los datos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién importado queda como proyecto activo en el sistema.

7.5. Tecnologías usadas

Las metas de la arquitectura que se buscaron fueron:

- Mínima acoplación y dependencia entre módulos
- Mayor facilidad en la persistencia
- La mayor separación entre lógica de negocios y presentación
- Separación de aspectos que no fueran parte del sistema propuesto
- Soporte por la mayor cantidad de navegadores
- Posibilidad de integración con otros sistemas

Dadas estas metas, se hizo una investigación acerca de las bibliotecas que permitieran alcanzar estas metas. De las posibilidades las bibliotecas elegidas fueron:

- Spring Framework de Inyección de dependencias, su soporte para Hibernate y a transacciones coordinadas.
- Spring Security para facilitar la implementación de las rutinas de seguridad en el sistema y permitir integrarse a diferentes formas de autenticación.

7. Objetivos de la herramienta

- Hibernate para facilitar la persistencia de los objetos.
- Struts2 para facilitar la realización de la interfaz web, separando la lógica de negocios de su presentación.
- AspectJ para separar en aspectos funcionalidad que complicaba el diseño de la herramienta.
- jQuery como biblioteca de JavaScript probada que soporta la mayor cantidad de navegadores.

Todas estas bibliotecas posibilitan programar sin atar los objetos a un framework en particular (POJO, plain old java object) permitiendo una flexibilidad y reusabilidad mucho mayor. [Ric06]

7.5.1. Inyección de dependencias

En ingeniería de software, un patrón de diseño es una solución general y repetible a un problema recurrente en el diseño de software. No es un diseño terminado que pueda ser transformado directamente en código, sino una descripción o plantilla de cómo resolver un problema que pueda ser usado en muchas diferentes situaciones. Los patrones de diseño orientados a objetos típicamente muestran relaciones e interacciones entre clases u objetos sin especificar que clases finales de la aplicación o que objetos se usarán. Parte de la utilidad de un patrón, es que no solamente describe la solución, sino también las consecuencias y concesiones que se tendrán que hacer al decidir aplicarlo.

En la comunidad Java, ha habido una profusión de contenedores ligeros que ayudan a ensamblar componentes de diferentes proyectos en una aplicación. Por debajo de estos contenedores hay un patrón común de cómo enlazar los componentes -conectar los cables- un concepto que se nombra bajo el nombre genérico de “inversión de control” (IoC). Aquí se profundiza como funciona este patrón bajo el nombre más específico de “inyección de dependencias”, y se compara contra la alternativa de los Service Locators (localizadores de servicio). La elección entre ellos es menos importante que el principio de separar configuración de uso.

Uno de los aspectos interesantes del mundo de java es la gran actividad construyendo alternativas a la tecnología J2EE, sobre todo en código abierto. Esto obedece a una reacción a la gran complejidad en el mundo del J2EE, pero también es para explorar alternativas. Un problema común es como juntar elementos dispares: como unir una arquitectura de controlador web con una interfaz de base de datos que fueron hechos por equipos diferentes con escaso conocimiento de los otros grupos. En este momento existen diferentes frameworks en escena y se están creando otros para proveer una capacidad general para ensamblar componentes de diferentes capas. Genéricamente se les nombra como contenedores ligeros, y como ejemplos están PicoContainer y Spring.

7.5.1.1. Componentes y servicios

El tema de unir los elementos entre sí, trae a la luz los temas recurrentes de terminología que rodean los términos servicios y componentes.

Componente significa un conjunto de software que fue hecho para ser usado sin cambios, por aplicaciones que están fuera del control de los escritores del componente. Por “sin cambio”, se refiere a que la aplicación que usa el componente no cambia el código del componente, aunque podría alterar su comportamiento extendiéndolo de formas permitidas por los autores del componente.

Un servicio es similar a un componente en que es usado por aplicaciones externas. La principal diferencia es que se espera usar un componente localmente (jar, assembly, dll, etc). Un servicio se usará remotamente por alguna interfaz remota, síncrona o asíncrona (web-service, sistema de mensajes, RPC o socket).

Esta situación se describe como un plugin. La clase implementadora del buscador no esta enlazada dentro del programa al momento de compilar, porque no se como vaya a ser usado. En vez de eso, se necesita que el listador trabaje con cualquier implementación, y que esa implementación se pueda

conectar después, fuera del control del desarrollador original. El problema es como se puede hacer ese enlace de manera que la clase listadora no conozca la clase implementadora, pero aún así pueda interactuar con una instancia para hacer su trabajo.

Hay tres aspectos importantes en la inyección de dependencias. Los nombres que se usan son: Constructor Injection, Setter Injection y Interface Injection. En las discusiones actuales acerca de Inversión de control, se refieren a IoC tipo 1 (interface injection), IoC tipo 2 (setter injection) y IoC tipo 3 (constructor injection) [Fow04].

PicoContainer usa un constructor para decidir como inyectar una implementación de un buscador en la clase listadora. Para que esto funcione, la clase listadora necesita declarar un constructor que incluya todo lo que necesita ser inyectado.

El framework Spring es un framework de amplio espectro para desarrollo de aplicaciones en Java enterprise edition. Incluye capas de abstracción para transacciones, frameworks de persistencia, aplicaciones web y JDBC. Al igual que PicoContainer soporta constructor injection y setter injection, pero sus desarrolladores prefieren usar setter injection.

La tercera forma de inyección es la que define y usa interfaces para la inyección, pero induce dependencias al framework específico que se esté usando y complica el código al tener que implementar una serie de interfaces en cada objeto que se necesite inyectar.

Si esta separación es útil en un solo proyecto de código, es vital cuando se están usando elementos ajenos tales como **componentes** y **servicios**. La primera cuestión es si se quiere diferir la elección de clase de implementación para instalaciones particulares. Si es el caso, se necesita alguna clase de implementación de plug-in. Una vez que se están usando plug-ins, es esencial que el ensamblador de los plug-ins sea hecha separadamente del resto de la aplicación, para que se puedan sustituir fácilmente diferentes configuraciones para diferentes instalaciones. Como lograrlo es secundario. Este mecanismo de configuración puede ser un localizador de servicio o usar inyección para configurar los objetos directamente.

7.5.2. Spring Framework

La misión de Spring framework es:

- Usar J2EE debería ser fácil.
- Es mejor programar orientado a interfaces que a clases. Spring reduce a cero la complejidad de usar interfaces.
- Usar JavaBeans es una forma de configurar aplicaciones.
- El diseño orientado a objetos es más importante que cualquier tecnología como J2EE.
- Se usan demasiado las checked exceptions en Java. Una plataforma no debería de forzar al usuario a cachar una excepción de la cual es poco probable que se pueda recuperar.
- La capacidad de probar el código es esencial, y una plataforma como Spring debería ayudar a hacer el código más fácil de probar.

Características Spring es una plataforma en capas para Java/J2EE basada en código publicado en Expert One-on-One J2EE Design and Development por Rod Johnson (Wrox, 2002)³. [Wal08]

Proporciona Entre su funcionalidad:

- El contenedor ligero más completo, dando una configuración y alambrado centralizado y automatizado de los objetos de aplicación. El contenedor no es invasivo y es capaz de ensamblar un

³<http://www.wrox.com/WileyCDA/WroxTitle/productCd-0764543857.html>

7. Objetivos de la herramienta

sistema completo a partir de un conjunto de componentes poco acoplados de forma consistente y transparente. Los componentes no necesitan heredar ni implementar interfaces específicas del framework ni del servidor de aplicación, convirtiéndolos en auténticos POJOs (plain old java objects). El contenedor provee agilidad y aumenta la capacidad de probar los componentes, permitiendo desarrollarlos y probarlos de forma aislada, luego integrarlos para instalarlos en cualquier entorno (J2SE o J2EE).

- Una capa de abstracción para el manejo de transacciones, permitiendo manejadores de transacciones intercambiables, y facilitando demarcar transacciones sin lidiar con cuestiones de bajo nivel. Incluye estrategias genéricas de JTA y una sola fuente de JDBC. Al contrario de usar directamente JTA o EJB CMT, el soporte transaccional de Spring no está atado a un entorno de J2EE.
- Una capa de abstracción de JDBC que ofrece una jerarquía coherente de excepciones, simplifica el manejo de errores y reduce drásticamente la cantidad de código que se necesita escribir.
- Integración con Toplink, Hibernate, JDO y iBATIS SQL Maps: en términos de recursos, implementación del soporte a DAO (data access objects) y estrategias transaccionales. El soporte a Hibernate integra muchas conveniencias de IoC, cubriendo muchos problemas de integración. Todo esto cumple con las jerarquías de Spring para transacciones genéricas y DAOs.
- Funcionalidad orientada a aspectos (AOP), totalmente integrada en la conflagración de Spring. Se puede habilitar con AOP cualquier objeto administrado por Spring, añadiendo aspectos como transacciones declarativas. Con Spring se pueden hacer transacciones declarativas sin EJB, incluso sin JTA si se esta usando una sola base de datos en Tomcat u otro container sin soporte a JTA.

Se puede usar toda la funcionalidad de Spring en cualquier servidor J2EE, y la mayoría en cualquier entorno no administrado. La meta principal es permitir objetos de negocio y de acceso reusables a datos que no estén atados a servicios J2EE específicos. Estos objetos se pueden reusar entre entornos J2EE (web o EJB), aplicaciones autónomas, entornos de prueba, etc. sin ningún problema [Joh03].

La arquitectura por capas de Spring permite mucha flexibilidad. Nueva funcionalidad usa la funcionalidad de los niveles inferiores.

La funcionalidad central de Spring puede ser usada por cualquier aplicación Java, pero hay extensiones para construir aplicaciones web en una plataforma Java Enterprise. Aunque Spring no impone ningún modelo de programación específico, se ha popularizado en la comunidad Java como una alternativa, un reemplazo o incluso una adición al modelo Enterprise JavaBeans (EJB) [Spr08].

7.5.3. Hibernate

La funcionalidad principal de Hibernate es mapear de clases de Java a tablas en bases de datos, y de tipos de Java a tipos de SQL, aunque también provee formas de interrogar y leer datos [BK07, hib]. Hibernate genera las sentencias de SQL y libera al desarrollador de manejar manualmente los resultados de la base de datos y la conversión de datos, manteniendo la aplicación portátil entre todas las bases de datos SQL soportadas, con una cuota de procesamiento extra muy pequeña.

La meta de Hibernate es liberar al desarrollador de hacer el 95 % de las tareas de programación para persistencia de datos, comparado con codificar manualmente usando SQL y JDBC.

Mapeo Mapear una clase de java a una tabla se logra con la configuración de un archivo XML o usando anotaciones Java. Al usar XML, Hibernate puede generar un esqueleto de código fuente de las clases persistentes, lo cual no es necesario si se usan anotaciones. Hibernate puede mantener un esquema de bases de datos a partir del XML o las anotaciones.

Existe funcionalidad para expresar relaciones uno a muchos y muchos a muchos entre clases. Adicionalmente a manejar las asociaciones entre objetos, Hibernate también puede manejar relaciones reflexivas donde un objeto tiene relaciones uno a muchos con instancias de su propio tipo.

Hibernate puede mapear tipos de usuario, lo que permite:

Cambiar el tipo por defecto que Hibernate elige al mapear una columna a una propiedad.

Mapear un enumerado de java a columnas como si fuera una propiedad regular.

Mapear una sola propiedad a múltiples columnas

Persistencia Hibernate provee persistencia transparente para cualquier objeto en Java (POJO, Plain Old Java Object). El único requisito es un constructor sin argumentos, no necesariamente público. En algunas aplicaciones también requiere poner atención en el comportamiento de los métodos equals() y hashCode().

Las colecciones de objetos se almacenan típicamente en objetos de Java Collections como Set y List. En la versión 5 de Java se pueden usar tipos genéricos. Hibernate se puede configurar para realizar carga postergada de las colecciones, lo cual es el comportamiento estándar en Hibernate 3.

Se puede configurar para que las operaciones se hagan en cascada en objetos relacionados. Esto puede reducir el tiempo de desarrollo y asegurar la integridad referencial. Una verificación de cambios (dirty checking) evita escrituras innecesarias a la base de datos al actualizar solo los campos modificados de los objetos persistentes.

Hibernate Query Language (HQL) Hibernate provee un lenguaje inspirado en SQL llamado Hibernate Query Language (HQL) que permite realizar en Hibernate búsquedas parecidas a SQL. Por otro lado se pueden usar objetos Criteria, como una alternativa a HQL.

7.5.4. Patrón de arquitectura MVC

El patrón de arquitectura modelo-vista-controlador (MVC, model-view-controller), permite aislar la lógica de negocio de asuntos de interfaz de usuario, resultando en una aplicación que es más fácil de modificar, tanto la apariencia visual de la aplicación o el modelo de negocios sin afectar el otro. En MVC, el modelo representa la información (los datos) de la aplicación, la vista corresponde a los elementos gráficos como texto, checkboxes, etc, y el controlador administra la comunicación entre los datos y las reglas de negocio usadas para manipular los datos del modelo.

Es común separar una aplicación en diferentes capas que corren en diferentes computadoras: la presentación/interfaz de usuario, la lógica de negocios y el acceso a datos. En MVC la capa de presentación está a su vez separada en modelo, vista y controlador.

Frecuentemente se ve MVC en aplicaciones web, donde la vista es la página HTML y el controlador es el código que reúne los datos dinámicamente y genera el contenido en el HTML. El modelo es el contenido, el cual normalmente se almacena en una base de datos o en XML, y las reglas de negocio que transforman el contenido basado en las acciones.

7.5.5. Apache Struts2

Apache Struts 2 es un framework para crear aplicaciones web empresariales. Está diseñado para acelerar todo el ciclo de desarrollo, desde la construcción, despliegue y el mantenimiento de la aplicación conforme pasa el tiempo [str].

Struts 2 esta construido basado en las mejores prácticas y usando patrones de diseño aprobados por la comunidad, lo cual también es cierto para la primera versión de Struts. De hecho una de las metas principales de Struts era incorporar el patrón MVC de las aplicaciones de escritorio en un framework para aplicaciones web. El patrón resultante se le llama patrón de modelo 2. Esto fue un factor crucial de éxito en la evolución de las aplicaciones web bien diseñadas, porque puso

7. Objetivos de la herramienta

la infraestructura para lograr fácilmente la separación de asuntos (separation of concerns) de MVC. Ésto permitió que desarrolladores con pocos recursos para semejantes lujos en la arquitectura usar una solución previamente hecha basada en mejores prácticas. Struts 1 es responsable por muchas de las aplicaciones web mejor diseñadas de los últimos 10 años. En algún punto, la comunidad se dio cuenta de las limitaciones del primer framework. Dada la vitalidad de la comunidad no fue difícil identificar los puntos frágiles o inflexibles del framework. Struts 2 saca ventaja de las lecciones aprendidas para tener una implementación de MVC más limpia, y al mismo tiempo introduce varias características de arquitectura que lo hacen más limpio y más flexible.

Estas nuevas características incluyen interceptores para separar en capas los cross-cutting concerns y separarlos de la lógica de las acciones, configuración basada en anotaciones para reducir o eliminar la configuración con XML, un poderoso lenguaje de expresiones, Object-Graph Navigation Language (OGNL, lenguaje para navegación en gráficas de objetos), que utiliza todo el framework, y un API de tags basado en un mini-MVC que soporta componentes de UI reusables y modificables.

7.5.6. AJAX

Ajax, o AJAX (Asynchronous JavaScript and XML, JavaScript y XML asíncrono) es un grupo de técnicas de desarrollo web interrelacionadas usadas para crear aplicaciones web interactivas [aja]. Con Ajax, las aplicaciones web pueden hacer peticiones de datos al servidor de forma asíncrona en segundo plano sin interferir con el despliegue y comportamiento de la página que se está viendo. El uso de Ajax ha posibilitado un aumento de la animación en páginas web. Los datos se recuperan usando el objeto XMLHttpRequest o scripts remotos en browsers que no lo soportan.

En el artículo que creó el término Ajax, Jesse James Garrett [Gar05] explicó que se requieren las siguientes tecnologías

- XHTML y CSS para la presentación
- El Document Object Model (DOM) para la presentación dinámica y la interacción con datos
- XML y XSLT para el intercambio, manipulación y despliegue de los datos
- El objeto XMLHttpRequest para la comunicación asíncrona
- JavaScript para juntar todas estas tecnologías

Sin embargo, desde entonces ha habido desarrollo en las tecnologías que se usan en las aplicaciones Ajax y en la definición del término Ajax.

- Javascript no es el único lenguaje de script que se puede usar para implementar una aplicación Ajax. Otros lenguajes como VBScript y el lenguaje EGL también son capaces de proporcionar la funcionalidad requerida.
- XML no se requiere para el intercambio de datos, y por lo tanto tampoco se requiere XSLT para la manipulación de los datos. Se usa frecuentemente JavaScript Object Notation (JSON) como un formato de intercambio de datos, aunque se pueden usar otros formatos como HTML preformado o texto plano.
- Las peticiones no tienen que ser asíncronas.

Por lo tanto se ha cambiado el acrónimo AJAX por Ajax, que no representa a estas tecnologías en específico.

7.5.7. Programación orientada a aspectos y AspectJ

La programación orientada a aspectos (Aspect Oriented Programming o AOP), es un paradigma de programación que incrementa la modularidad al mejorar la separación de ciertas secciones de la aplicación. Esto conlleva dividir un programa en distintas partes (concerns). Todos los paradigmas de programación aportan algún nivel de agrupamiento y encapsulación de concerns (secciones) en entidades separadas e independientes al proveer abstracciones (procedimientos, módulos, clases, métodos) que se pueden usar para implementar, abstraer y hacer composiciones de estas secciones. Pero algunos concerns no se pueden separar con estas formas de implementación (cross-cutting concerns) y se encuentran presentes en varias partes de la aplicación porque “cortan” a través de las secciones del programa. [aop]

La programación orientada a aspectos es una forma de modularizar funcionalidad que rompe la modularidad de la misma forma que la programación orientada a objetos es una forma de modularizar funcionalidad relacionada.

Un ejemplo de un cross-cutting concern son las bitácoras porque para implementar las bitácoras, éstas necesariamente afectan cada parte del sistema, porque las bitácoras están presentes en todas las clases y métodos.

Todas las implementaciones de AOP tienen algunas expresiones que encapsulan cada concern en un lugar. La diferencia entre las implementaciones radica en el poder, seguridad y usabilidad de las herramientas disponibles. Por ejemplo, los interceptores que especifican los métodos a interceptar expresan una forma limitada de crosscutting, casi sin soporte para la verificación de tipos o para debuggear. AspectJ es una implementación de programación orientada a aspectos para Java. [aspa, aspb]

Uno de los elementos más importantes del diseño de un lenguaje orientado a aspectos, es el modelo de join point. Este modelo provee la base sobre la cual es posible definir la estructura dinámica de los crosscutting concerns.

Cada llamada a un método durante la ejecución, es un join point diferente, aún si viene de la misma expresión en el programa. Se pueden ejecutar muchos otros join points mientras se ejecuta un join point de llamada a un método, todos los join points que se invocan mientras se ejecuta el cuerpo del método, y de todos los métodos llamados desde éste.

Pointcuts Los pointcuts permiten a los programadores especificar join points (momentos bien definidos en la ejecución de un programa, como una llamada a método, instanciación de un objeto o el acceso a una variable). Todos los pointcuts son expresiones (cuantificaciones) que determinan cuando un determinado join point coincide.

Advice De esta forma, los pointcuts eligen los join points. Pero no hace mucho más que calificarlos. Para que se realice algún comportamiento, se usa el advice. El advice junta un pointcut (con sus join points) y un código que se ejecutará en cada uno de los join points.

AspectJ tiene diferentes tipos de advice. El advice de tipo *before* (antes) se ejecuta cuando se alcanza un join point, antes de que el programa ejecute el join point. Por ejemplo, un *before advice* en una llamada a un método corre antes del que el método se ejecute, justo después de que se evalúen los argumentos de la llamada.

El advice de tipo *after* (después) en un join point específico corre después de que el programa ejecutó el join point. Por ejemplo un *after advice* en una llamada a un método corre después de que se ha ejecutado completo, justo antes de que se regrese el control al método que lo llamó. Dado que los programas en Java pueden terminar “normalmente” o lanzar una excepción, hay tres tipos de advice: *after returning* (después de retornar normalmente), *after throwing* (después de retornar lanzando una excepción), y *after* (que corre después de retornar normalmente o con excepción).

7. Objetivos de la herramienta

Finalmente el *around advice* (alrededor) en un join point corre en cuanto se alcanza el join point, y continúa corriendo después de que se termina la ejecución del join point. A diferencia de los otros, tiene control explícito si el programa procede o no a ejecutar el join point.

Declaraciones Inter-tipos Las declaraciones Inter-tipos (inter-type) en AspectJ son declaraciones que cortan alrededor de las clases y sus jerarquías. Pueden declarar miembros que cortan múltiples clases, o cambiar las relaciones de herencia entre varias clases. A diferencia del advice que opera principalmente dinámicamente, las introducciones operan estáticamente en tiempo de compilación.

Considérese el problema de expresar una funcionalidad compartida por algunas clases existentes que ya son parte de una jerarquía de clases, por ejemplo, ya extienden una clase. En Java, se crea una interfaz que captura esta nueva funcionalidad y se le puede agregar a las clases necesarias un método que implementa esta interfaz.

De esta forma AspectJ puede expresar la funcionalidad en sólo un sitio, al usar declaraciones intertipo. El aspecto declara los campos y métodos que son necesarios para implementar esta nueva capacidad, y asociar los métodos y campos a las clases existentes.

Aspectos Los aspectos engloban los pointcuts, advice y declaraciones intertipo en una unidad modular de implementación. Se definen definir de una forma muy similar a una clase, y pueden tener métodos, campos, inicializadores adicionalmente a los miembros que cortan la funcionalidad. Dado que sólo los aspectos pueden tener miembros que cortan funcionalidad, este tipo de declaraciones están concentradas.

Igual que las clases, los aspectos pueden instanciarse, pero AspectJ controla como se realiza la instanciación, de forma que no se puede usar el operador `new` de Java para construir nuevas instancias de un aspecto. De forma predeterminada, un aspecto es un singleton de forma que sólo existe una instancia de éste en todo el sistema. Gracias a esto, los advice pueden utilizar campos no estáticos del aspecto.

7.5.8. Sistemas basados en el conocimiento

Los programas tradicionales pueden ser divididos en:

- Un algoritmo, en el cual está implícito todo el conocimiento.
- Datos, a los cuales se les aplica el algoritmo.

Estos programas sólo pueden dar respuestas a problemas para los cuales fueron específicamente programados. Si el programa necesita ser modificado para incluir nueva información, debe ser examinado por completo, y para cumplir el fin para el cual fue programado, debe de ejecutar una secuencia predeterminada de instrucciones.

Un Sistema Basado en el Conocimiento (SBK) es un conjunto de programas de computo ligado a una base de conocimientos adquirida generalmente a través de varios expertos humanos dentro de un dominio específico (la nube) que incluye un conjunto de algoritmos, reglas o técnicas de razonamiento a través de los cuales se pueden hacer inferencias para la solución de problemas o que permiten dar recomendaciones para el análisis, el diagnóstico, apoyar a técnicas de enseñanza y/o en general recomendar, actuar y explicar sus acciones en actividades en las cuales se requiere generalmente del saber de expertos humanos dentro de una nube específica. [Ser08]

Un sistema basado en el conocimiento independiza los procedimientos para la solución de problemas de los datos respectivos. Las modificaciones se realizan alterando la base de conocimientos sin afectar la estructura del programa completo. Estos sistemas seleccionan los medios y hechos para obtener una respuesta adecuada a una situación específica.

El mecanismo de razonamiento, también llamado máquina de inferencia o estructura de control, es un conjunto de rutinas para el control de decisiones así como para el manejo de preguntas al usuario. En esta parte del sistema basado en el conocimiento no debería contener, información acerca del dominio específico del problema y debe contar con cierta independencia para permitir un grado de generalidad.

A pesar de lo anterior, la base de conocimientos y el mecanismo de razonamiento están íntimamente ligados, dado que la primera presenta la organización del conocimiento en base al cual es segundo va a trabajar. Existe entonces una estrecha relación entre la organización del sistema basado en el conocimiento y su control.

Los mecanismos de razonamiento no pueden ser completamente independientes del tipo de problemas a resolver, sin embargo, estas rutinas de razonamiento preferentemente no deben de ser tan especializadas, pues no podrían entonces aplicarse a otro tipo de problemas.

El razonamiento del sistema basado en el conocimiento debe ser guiado de alguna manera con el fin de proporcionar una secuencia de respuestas válidas y coherentes al usuarios en un medio ambiente aceptable. Para tal guía es necesario un conjunto de estrategias de control, las cuales son:

- Estrategias de razonamiento
- Estrategias de explicación
- Estrategias de interpretación

El poder de un sistema basado en el conocimiento podría medirse de acuerdo a que tan correctamente son aplicadas las técnicas de razonamiento sobre el conocimiento almacenado. El éxito del sistema radicará en encontrar una buena respuesta a un problema en base a los recursos con que se disponga. La eficiencia en la búsqueda de soluciones afecta directamente al éxito del sistema.

7.5.8.1. Arquitectura

A pesar de la diferencia de enfoques los sistemas basados en el conocimiento genéricamente se puede decir que tienen los siguientes componentes.

Base de conocimiento la cual contiene el saber específico en la disciplina en la que se enfoca el sistema. Consiste en un conjunto de hechos (datos) y de reglas programadas, sin que contenga información de un problema en particular. El conocimiento plasmado en la base juega el papel más importante en la calidad y habilidad del sistema.

Interfaz con el usuario la cual permite aceptar y reconocer un lenguaje de comandos de forma natural y los traduce en instrucciones y datos para que el sistema basado en el conocimiento trabaje.

Interfaz con el experto la cual permite captar información del exterior (proporcionada por un experto) e introducirla en forma adecuada en la base de conocimiento.

Mecanismo de razonamiento o máquina de inferencia que permite controlar el sistema. Maneja las reglas contenidas en la base de conocimiento.

Espacio de trabajo lugar en donde se almacena los resultados, hipótesis y decisiones intermedias, así como el estado del programa.

7.5.9. Drools

Drools es un sistema de manejo de reglas (BRMS) que usa un motor de inferencia de forward chaining basado en reglas, llamado un sistema de reglas de producción que usa una implementación del algoritmo de Rete, de Charles Forgy. [dro]

El forward chaining es uno de los dos métodos principales al usar reglas de inferencia en inteligencia artificial, y en la lógica se le refiere como “modus ponens”. [for]

El forward chaining comienza con los datos disponibles y usa las reglas de inferencia para extraer más datos hasta que se alcance una meta. Un motor de inferencia que use forward chaining busca las reglas de inferencia hasta que encuentra una en el que el antecedente sea verdadero. Al encontrarlo, puede concluir el consecuente, adicionándolo esta información a la previamente existente. La máquina seguirá realizando el proceso de forma iterativa hasta que se alcance una meta.

Dado que los datos determinan que reglas se seleccionan y usan, este método se le denomina “dirigido por los datos”, en contraste a un sistema “dirigido por las metas” que use backward chaining. El enfoque de forward chaining se usa frecuentemente en los sistemas basados en el conocimiento, como CLIPS.

Una de las ventajas del forward chaining sobre backward chaining es que la recepción de nueva información puede disparar nuevas inferencias, lo cual hace que el motor esté mejor adaptado a situaciones dinámicas en donde las condiciones cambian.

7.6. Arquitectura

El sistema propuesto y realizado como parte de esta tesis contiene los siguientes módulos.

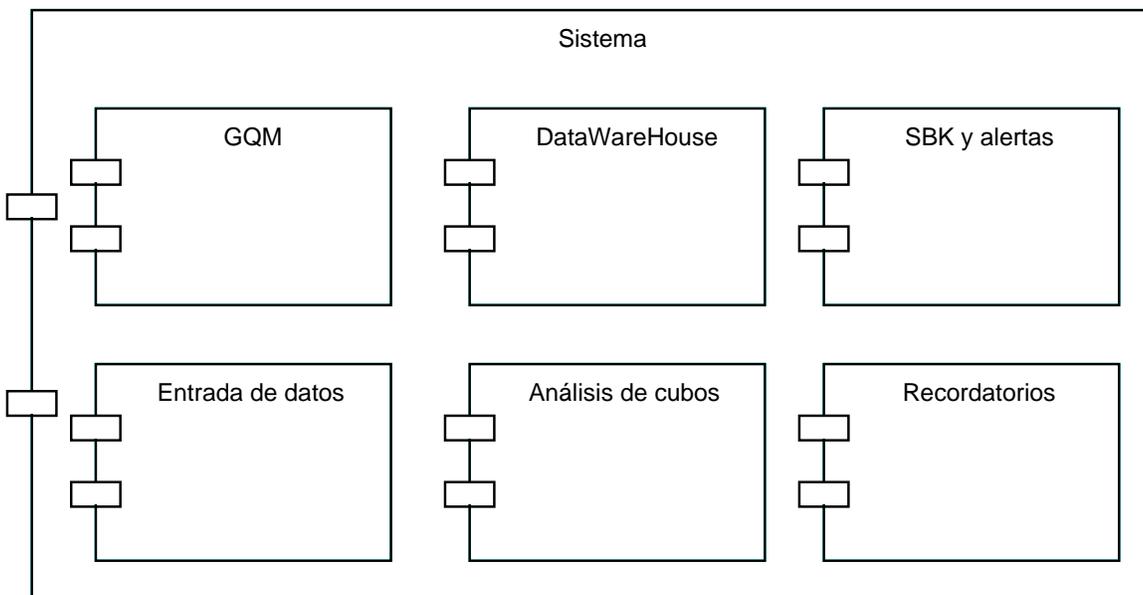


Figura 7.3.: Diagrama de componentes del sistema

GQM Este módulo se enfoca a la planeación de la medición, que medir, como medirlo, porque medirlo, cuando medirlo, quien lo tiene que medir...

Entrada de Datos Este módulo se enfoca a recabar los datos que se han definido en el módulo anterior y los guarda en el sistema.

Data Warehouse Este módulo permite almacenar los datos del sistema en un formato fácilmente explotable.

Análisis y dashboard/reportes En este módulo se procesan de forma flexible datos existentes en el sistema para producir nuevos datos.

También permite consultar la información contenida en el sistema y permite que los usuarios la visualicen de diferentes formas.

Recordatorios Este módulo permite definir avisos a los usuarios para que llenen los formularios en intervalos programados.

SBK Este módulo permitirá definir comportamientos disparados por el análisis por reglas de inferencia de los datos, como por ejemplo alertas y consejos al usuario. Ver la sección 7.5.8 en la página 66.

7.7. Diagrama de clases

Para el modelado de la arquitectura se optó por los Diagramas de Clases de UML. Éstos muestran las distintas clases de las que dispone el sistema y sus relaciones.

La arquitectura del sistema fue dividido en tres capas, de acuerdo a la función que las clases representan:

- Acceso a datos

- Modelo de datos y reglas de negocio

- Capa de interfaz de usuario

7.7.1. Modelo de datos

En el diagrama de clases del modelo de datos se plasman las clases que se usaron para la capa de objetos de negocio de la aplicación.

El tejido de las clases es bastante denso en enlaces internos, pero no tienen relaciones con clases externas al modelo de datos.

7. Objetivos de la herramienta

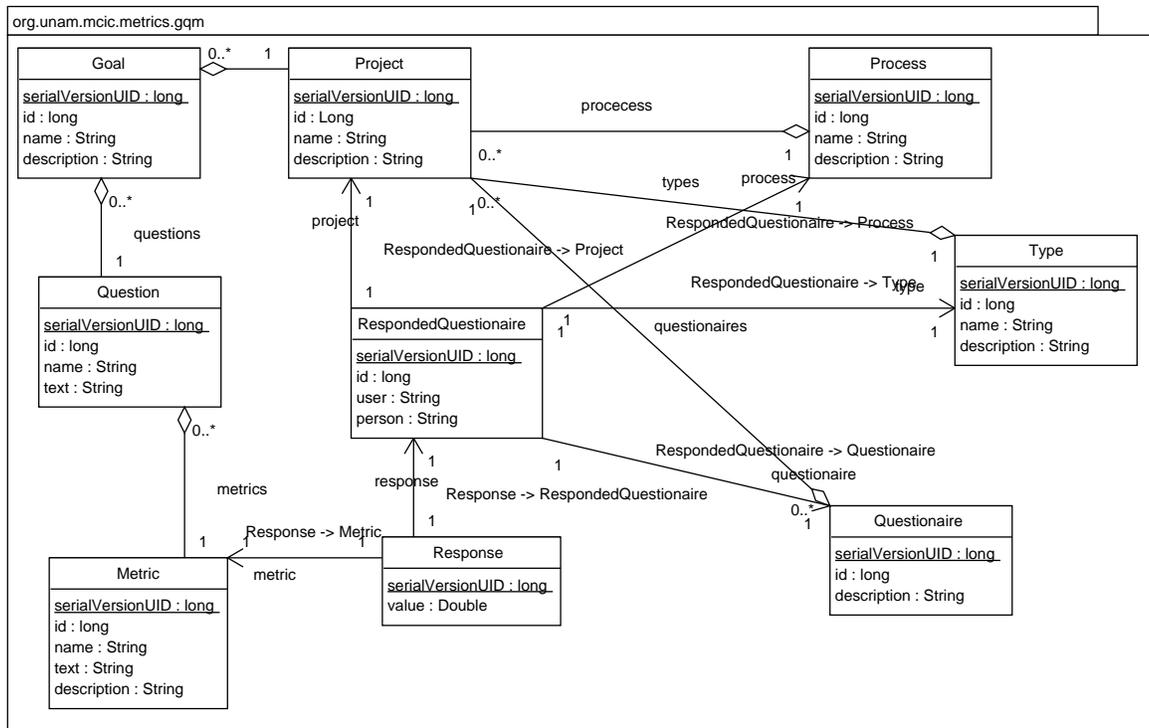


Figura 7.4.: Diagrama de clases del modelo de datos

7.7.2. Acceso a datos

La capa de acceso a datos provee una abstracción de la persistencia de los datos del sistema. Su razón es proveer una abstracción del acceso a datos para hacer más modular el sistema.

Su diseño permite reemplazarla fácilmente con otra tecnología de persistencia diferente a Hibernate. La clave del diseño es que las clases de interfaz de usuario están diseñadas a partir de interfaces (según la noción de Java) que proporcionan las operaciones necesarias para la operación del sistema, lo cual permite que los métodos de servicio de las clases concretas permanezcan ocultos, y por lo tanto, mientras las clases se adhieran a las interfaces, pueden tener métodos extras, incluso formas de creación especiales, y aún así puedan ser inyectadas exitosamente a las clases que dependen de las interfaces.

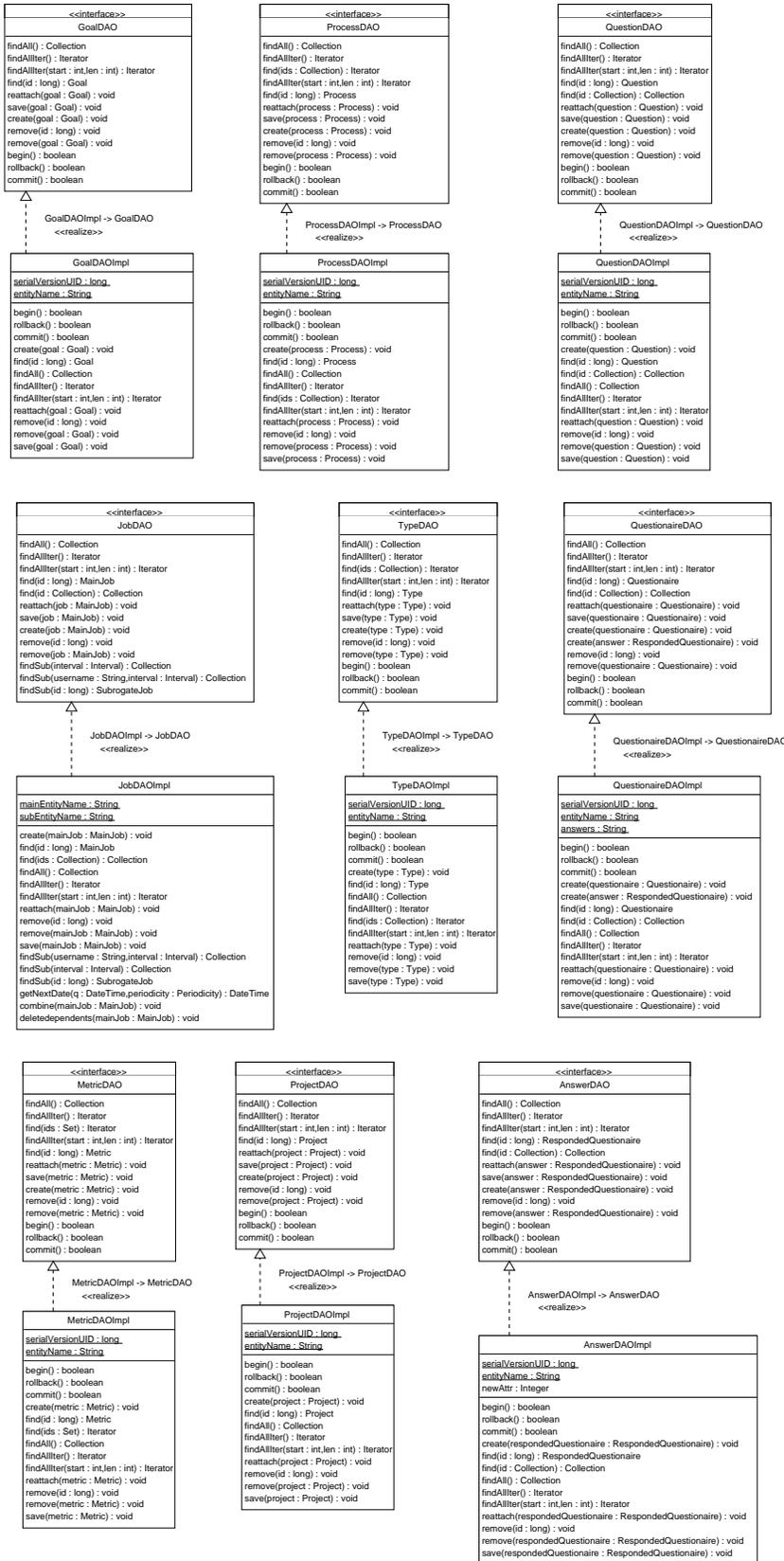


Figura 7.5.: Diagrama de clases de Acceso a datos

7. Objetivos de la herramienta

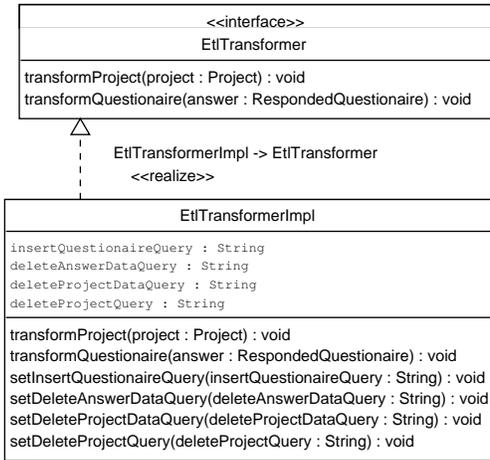


Figura 7.6.: Diagrama de clases de ETL

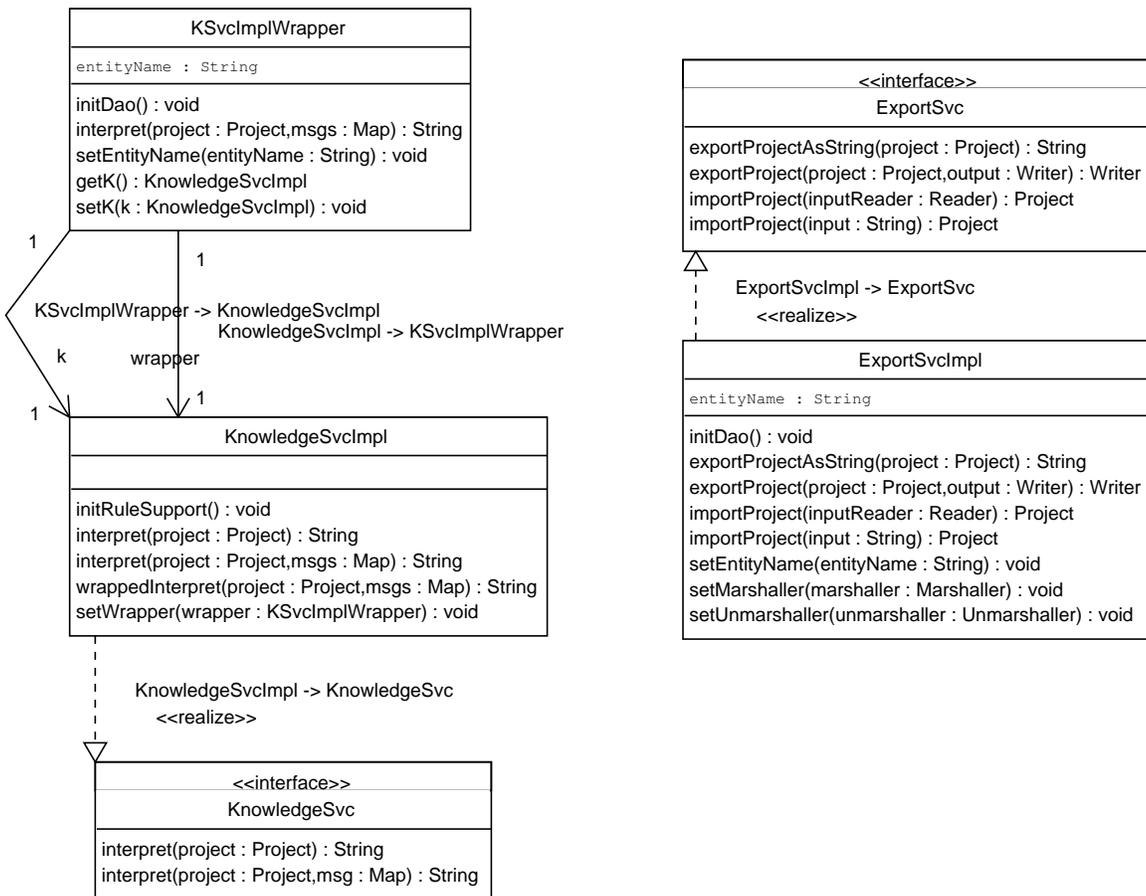


Figura 7.7.: Diagrama de clases de servicios

7.7.3. Capa de Interfaz de usuario

Finalmente la capa de interfaz de usuario, da la funcionalidad para poder visualizar e interactuar con el sistema. A continuación se muestran dos ejemplos.

En la clase de la figura 7.8 en la página siguiente se observan atributos que representan el estado de

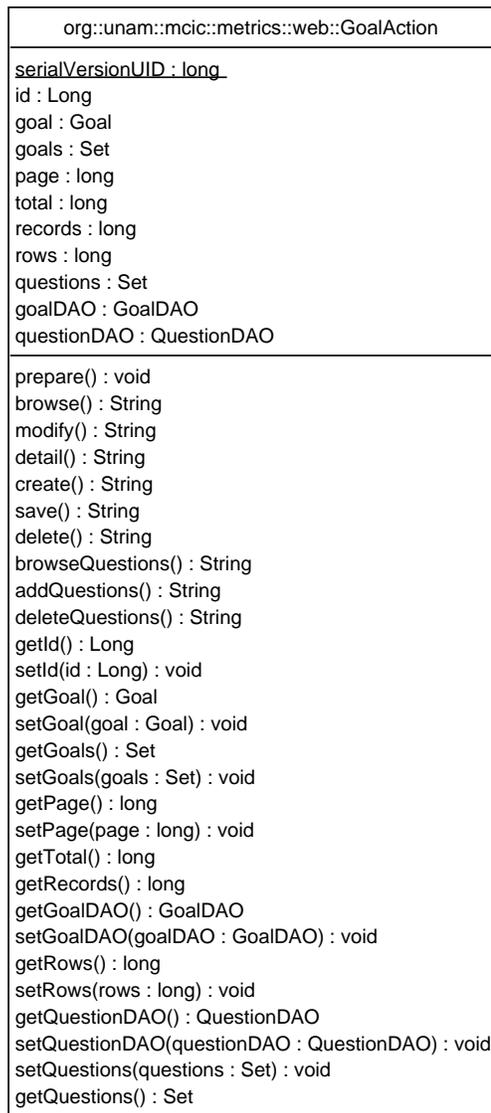


Figure 7.8.: Capa de interfaz: GoalAction

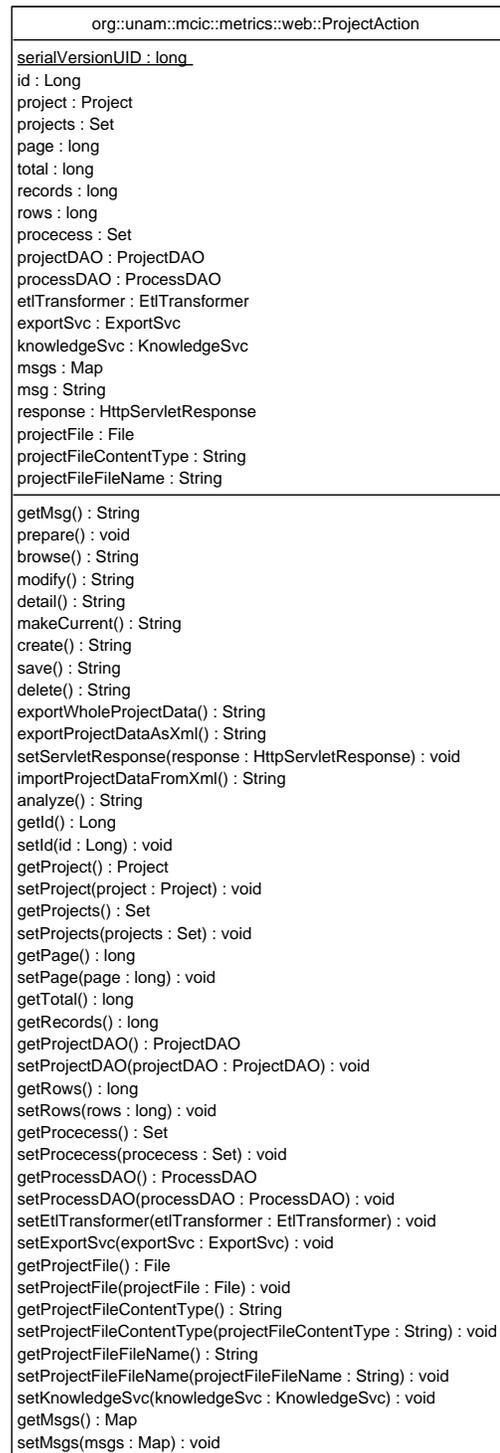


Figure 7.9.: Capa de interfaz: ProjectAction

7. Objetivos de la herramienta

una invocación y métodos para las distintas funcionalidades como ver un listado de metas, ver el detalle de una meta, crear una nueva meta, salvarla, modificarla, borrarla, así como métodos internos de Struts (prepare), métodos para inyectar las clases de servicio (`getQuestionDAO()`, `setQuestionDAO()`) y getters y setters necesarios para mantener el estado del sistema entre invocaciones (`getGoal()`, `setGoal()`).

En la clase `ProjectAction` en la figura 7.8 en la página anterior, se observa el mismo diseño de la clase anterior. A excepción del método `prepare()` todos los demás métodos no tienen ninguna relación con Struts2, lo que convierte a esta clase en casi un auténtico POJO.

Todos los métodos de acción regresan una cadena, que después usa el controlador de Struts2 para buscar cual es la vista (jsp, freemaker, tiles, jFreeChart, JSON, XML, File, etc) que debe invocar. Ésto se define en un archivo de configuración XML fuera de las clases, de forma que la vista se puede alterar totalmente independientemente de la lógica de negocios o de presentación.

Los nombres de los atributos de las clases de servicio corresponden con los nombres especificados en la configuración de Spring, y por lo tanto no es necesario configurar explícitamente qué clases de servicio necesitan ser inyectadas en cada bean de Struts (autowiring by name) siempre y cuando sean de un tipo compatible. También se puede usar el tipo (autowire by type), pero éste enfoque tiene problemas cuando hay más de una clase de servicio con el mismo tipo, pero diferente nombre.

7.7.3.1. La recolección de eventos de calendario y los aspectos

Debido a que la recolección de eventos de calendario, por ejemplo, cuando un usuario llena un cuestionario a través de una notificación que recibió, es un reto especial. Dado que el sistema tiene la capacidad de realizar diferentes tipos de notificaciones, y que las notificaciones deben de cambiar de estado de “no realizadas” a “realizadas” sólo hasta que el usuario efectivamente la ha realizado, por ejemplo, hasta que llena un cuestionario exitosamente, bajo un diseño tradicional hubiera sido necesario poner el código de anulación de notificación en varios sitios del sistema.

Esta es uno de las situaciones para las que los aspectos están pensados. En lugar de colocar el código necesario para desactivar las notificaciones en cada clase que tenga que reconocer y desactivar notificaciones, y dado que no son parte del objetivo primario de la clase, se diseñó un aspecto que permitió integrar lo necesario al código de varias de las clases de la capa de interfaz, sin tocar el código principal. Dado que el framework de interfaz (struts) usa reflexión de Java, sólo fué necesario los campos adicionales necesarios a las clases que reciben la información de la notificación que se está respondiendo, y se envolvieron los métodos originales para que realicen el cambio de estado de la notificación para completar la funcionalidad.

Con base en los listados de código 7.1 en la página siguiente y 7.2 en la página 76, explicaré cómo fue realizado lo anterior.

Con `private String ResponseAction.refererId` y `private DateTime ResponseAction.dateProgrammed` se agrega un nuevo miembro a la clase `ResponseAction`. Estos miembros no son accesibles desde el código base de la clase en sí, pero lo son desde el aspecto. Su visibilidad esta definida como privada, así es que tampoco es posible accederlos desde otras clases.

Para poder acceder a los miembros que se acaban de agregar a la clase, se agregaron getters y setter, que son accesibles binariamente mediante las funciones `public DateTime ResponseAction.getDateProgrammed()`, `public void ResponseAction.setDateProgrammed(DateTime dateProgrammed)`, `public String ResponseAction.getRefererId()` y `public void ResponseAction.setRefererId(String refererId)`. Estas son las funciones que usa Struts para poder introducir los valores de la notificación.

Finalmente la funcionalidad se encapsula en el advice `after(ResponseAction r) returning(String ret) : ResponseActionSaveP(r)` que procesa la notificación cuando la expresión del point cut `pointcut ResponseActionSaveP (ResponseAction r): execution(String ResponseAction.save()) && target (r)` coincide. El pointcut se activa cuando se ejecuta la función `ResponseAction.save()` y el advice se ejecuta después de que la función retorna sin error.

Algorithm 7.1 Código de ResponseAction.java

```

public class ResponseAction extends ActionSupport implements Preparable,Parameterizable {
    private static final long serialVersionUID = -5616676083430715168L;
    protected String id;
    protected String projectId;
    protected Questionnaire questionnaire;
    protected Set<Questionnaire> questionnaires;
    protected Set<Long> metrics = new HashSet<Long>();
    protected Project project;
    protected RespondedQuestionnaire respondedQuestionnaire;
    protected String proc;
    protected String type;
    protected ProjectDAO projectDAO;
    protected ProcessDAO processDAO;
    protected QuestionnaireDAO questionnaireDAO;
    protected MetricDAO metricDAO;
    protected TypeDAO typeDAO;
    protected HibernateUserDetailsService userSvc;
    protected Map<String,Double> resp = new HashMap<String,Double>();
    protected Map<String, String[]> request;
    ...
    public String save(){
        for (String k:request.keySet()){
            if(k.startsWith("response")){
                String [] vt = request.get(k);
                Double v = Double.parseDouble(vt[0]);
                resp.put(k.substring(k.indexOf('[')+1, k.indexOf(']')), v);
            }
        }
        if(respondedQuestionnaire !=null && respondedQuestionnaire.fillAnswers( resp )){
            respondedQuestionnaire.setProcess(processDAO.find(proc));
            respondedQuestionnaire.setType (typeDAO.find(type) );
            respondedQuestionnaire.setUser (userSvc.getCurrentUserName() );
            respondedQuestionnaire.setDateCollected(TimeUtils.now());
            if(projectId!=null)
                projectDAO.find(projectId).addRespondedQuestionnaire(respondedQuestionnaire);
            else
                project.addRespondedQuestionnaire(respondedQuestionnaire);
            questionnaireDAO.create( respondedQuestionnaire );
            return SUCCESS;
        }
        return ERROR;
    }
    ...
}

```

Algorithm 7.2 Código de RefererAspect.aj

```

public aspect RefererAspect implements InitializingBean {
    protected int days = 15;
    protected JobDAO jobDAO;
    public int getDays() {
        return days;
    }
    public void setDays(int days) {
        this.days = days;
    }
    public void setJobDAO(JobDAO jobDAO) {
        this.jobDAO = jobDAO;
    }
    private String ResponseAction.refererId ;
    public String ResponseAction.getRefererId() {
        return refererId;}
    public void ResponseAction.setRefererId(String refererId) {
        this.refererId = refererId;}
    private DateTime ResponseAction.dateProgrammed;
    public DateTime ResponseAction.getDateProgrammed() {
        return dateProgrammed;}
    public void ResponseAction.setDateProgrammed(DateTime dateProgrammed) {
        this.dateProgrammed = dateProgrammed;}
    public Date ResponseAction.getDateProgrammedAsDate() {
        return dateProgrammed.toDate();}
    public void ResponseAction.setDateProgrammedAsDate(Date dateProgrammed) {
        this.dateProgrammed = new DateTime(dateProgrammed); }

    pointcut ResponseActionSaveP (ResponseAction r):
        execution(String ResponseAction.save())
        && target (r);
    after(ResponseAction r) returning(String ret) : ResponseActionSaveP(r) {
        if (ret.equalsIgnoreCase("SUCCESS"))
            if ( r.refererId != null ){
                SubrogateJob s = jobDAO.findSub( r.refererId );
                if (s!=null){
                    s.setDone(r.respondedQuestionnaire.getDateCollected());
                    if (r.getDateProgrammed() !=null && r.respondedQuestionnaire.getDateCollected().is
                        s.setStatus(SubrogateJob.Status.outdated);
                }
                else
                    s.setStatus(SubrogateJob.Status.filled);
            }
        }
    }
}

```

7. Objetivos de la herramienta

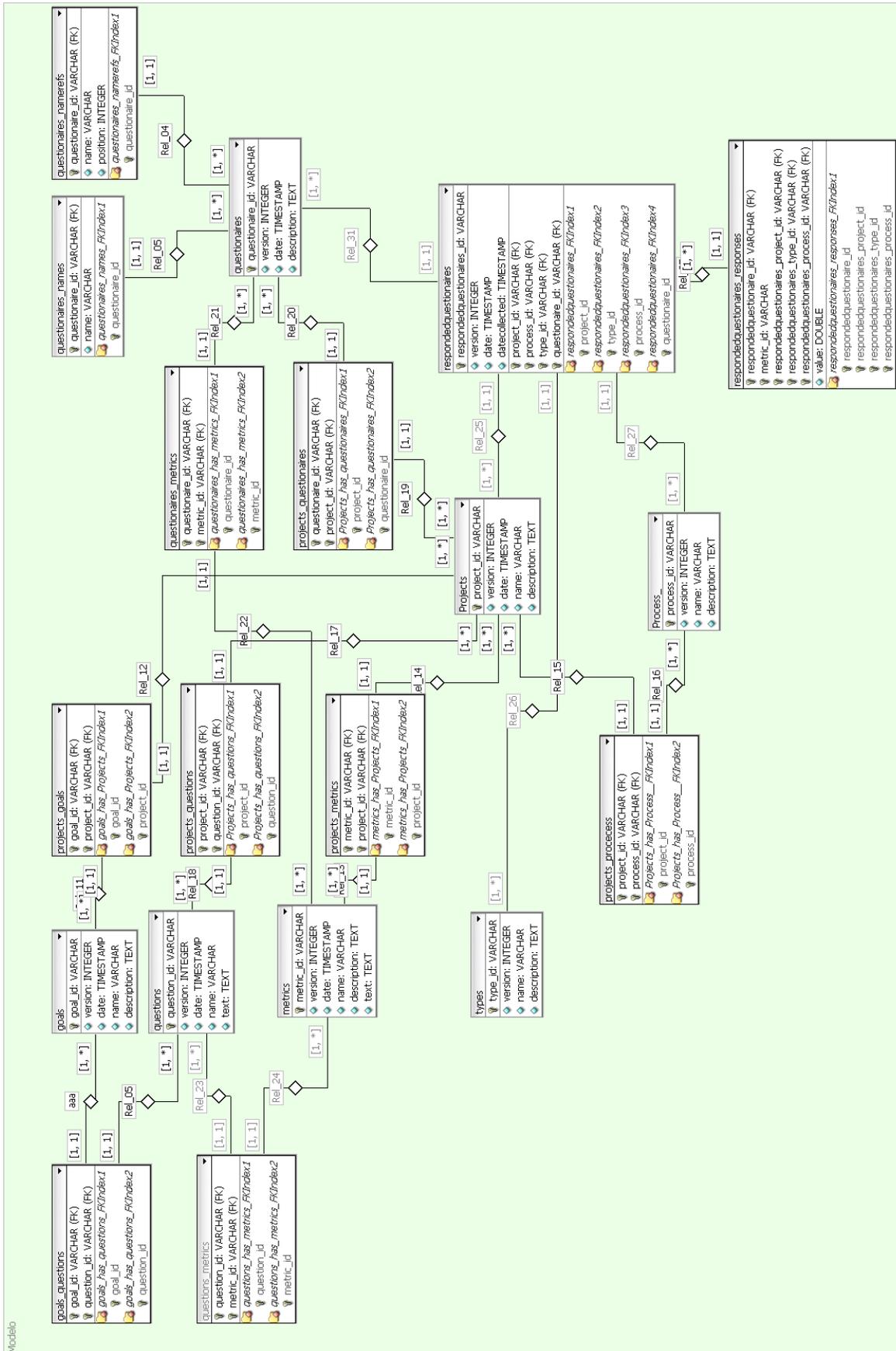


Figura 7.10.: Diagrama de base de datos

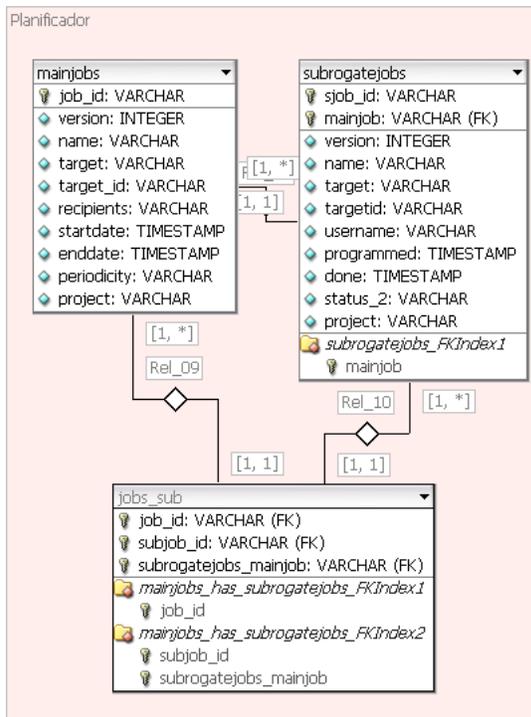


Figura 7.11.: Diagrama de base de datos del planificador

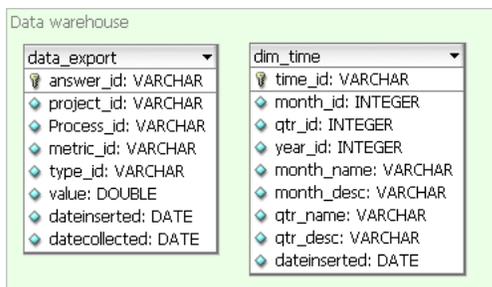


Figura 7.12.: Diagrama de base de datos del data warehouse

7. Objetivos de la herramienta

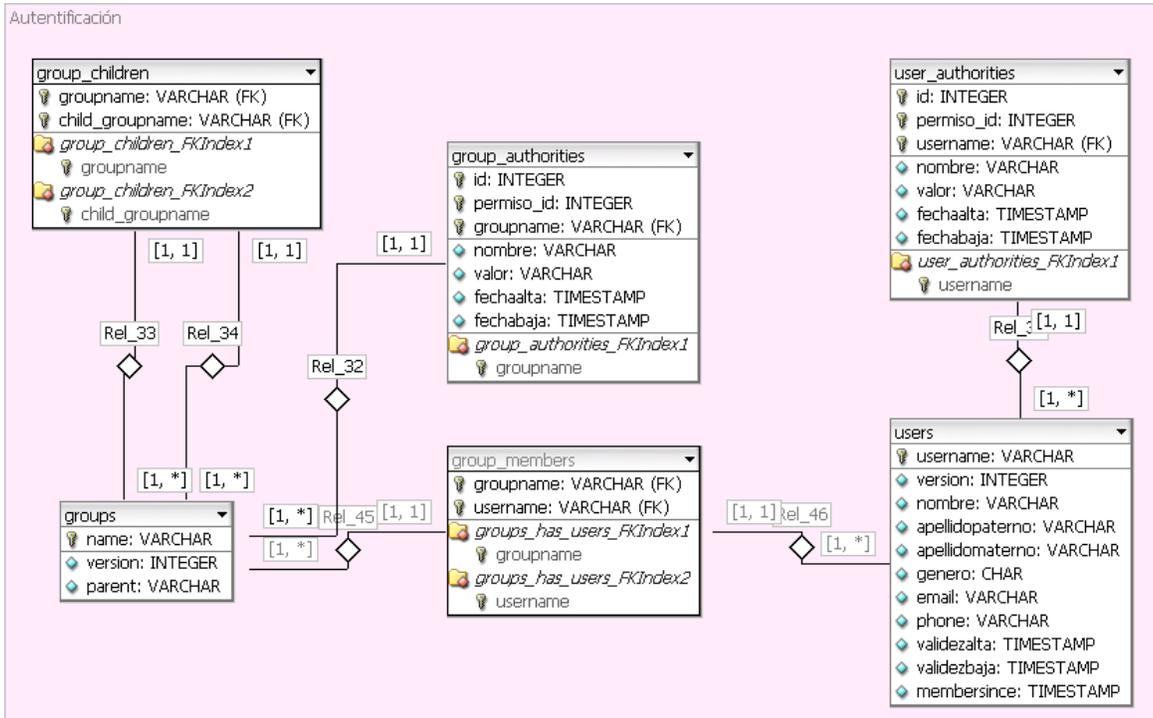


Figura 7.13.: Diagrama de base de datos de autenticación

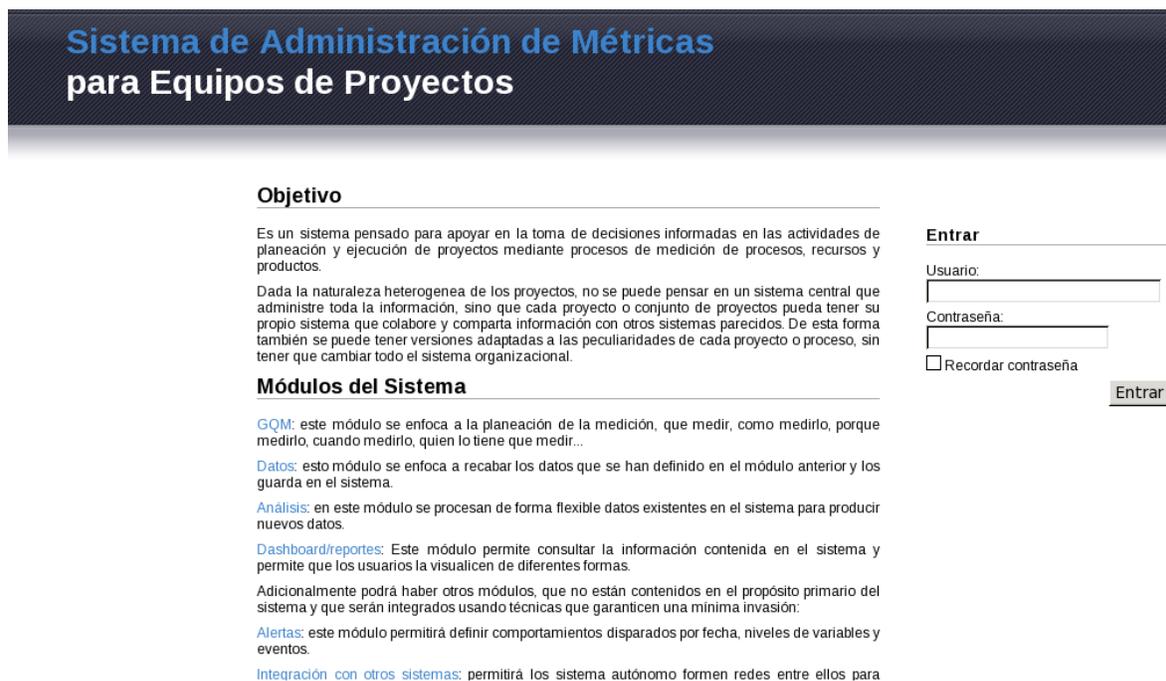
Después de haber hablado de los objetivos y la arquitectura de la herramienta propuesta, así como un ambiente en el que tendría mayor impacto positivo, en el siguiente capítulo se incluye un manual a alto nivel de la herramienta. En el apéndice 1 se pueden consultar el resto de los casos de uso que componen la descripción de la funcionalidad de la herramienta.

8. Manual de usuario

El siguiente manual de usuario está basado en la presentación y funcionalidad que tiene el sistema en un dispositivo tipo PC. En el apéndice ?? se puede apreciar la herramienta usada desde algunos dispositivos móviles.

8.1. Pantalla de Bienvenida

Al abrir el sitio web del sistema, el usuario se encuentra con la pantalla 8.1, la cual explica las posibilidades y alcances del sistema, y permite el ingreso del usuario al proporcionar sus credenciales.



Sistema de Administración de Métricas para Equipos de Proyectos

Objetivo

Es un sistema pensado para apoyar en la toma de decisiones informadas en las actividades de planeación y ejecución de proyectos mediante procesos de medición de procesos, recursos y productos.

Dada la naturaleza heterogénea de los proyectos, no se puede pensar en un sistema central que administre toda la información, sino que cada proyecto o conjunto de proyectos pueda tener su propio sistema que colabore y comparta información con otros sistemas parecidos. De esta forma también se puede tener versiones adaptadas a las peculiaridades de cada proyecto o proceso, sin tener que cambiar todo el sistema organizacional.

Módulos del Sistema

GQM: este módulo se enfoca a la planeación de la medición, que medir, como medirlo, porque medirlo, cuando medirlo, quien lo tiene que medir...

Datos: este módulo se enfoca a recabar los datos que se han definido en el módulo anterior y los guarda en el sistema.

Análisis: en este módulo se procesan de forma flexible datos existentes en el sistema para producir nuevos datos.

Dashboard/reportes: Este módulo permite consultar la información contenida en el sistema y permite que los usuarios la visualicen de diferentes formas.

Adicionalmente podrá haber otros módulos, que no están contenidos en el propósito primario del sistema y que serán integrados usando técnicas que garanticen una mínima invasión.

Alertas: este módulo permitirá definir comportamientos disparados por fecha, niveles de variables y eventos.

Integración con otros sistemas: permitirá los sistema autónomo formen redes entre ellos para

Entrar

Usuario:

Contraseña:

Recordar contraseña

Entrar

Figura 8.1.: Pantalla de Bienvenida

8.2. Entrada al sistema

El usuario proporciona unas credenciales (usuario y password u otra forma de autenticación) autorizadas y es presentado con la pantalla 8.2, que contiene el menú principal a la izquierda, y el área de trabajo a la derecha, conteniendo el mismo texto que la pantalla de bienvenida. Dado que el usuario acaba de iniciar su sesión y no ha seleccionado un proyecto, el espacio aparece vacío y se le invita al usuario a seleccionar un proyecto.

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Seleccione un proyecto] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas

Objetivo

Es un sistema pensado para apoyar en la toma de decisiones informadas en las actividades de planeación y ejecución de proyectos mediante procesos de medición de procesos, recursos y productos.

Dada la naturaleza heterogénea de los proyectos, no se puede pensar en un sistema central que administre toda la información, sino que cada proyecto o conjunto de proyectos pueda tener su propio sistema que colabore y comparta información con otros sistemas parecidos. De esta forma también se puede tener versiones adaptadas a las peculiaridades de cada proyecto o proceso, sin tener que cambiar todo el sistema organizacional.

Módulos del Sistema

GQM: este módulo se enfoca a la planeación de la medición, que medir, como medirlo, porque medirlo, cuando medirlo, quien lo tiene que medir...

Datos: este módulo se enfoca a recabar los datos que se han definido en el módulo anterior y los guarda en el sistema.

Análisis: en este módulo se procesan de forma flexible datos existentes en el sistema para producir nuevos datos.

Dashboard/reportes: Este módulo permite consultar la información contenida en el sistema y permite que los usuarios la visualicen de diferentes formas.

Adicionalmente podrá haber otros módulos, que no están contenidos en el propósito primario del sistema y que serán integrados usando técnicas que garanticen una mínima invasión:

Alertas: este módulo permitirá definir comportamientos disparados por fecha, niveles de variables y eventos.

Integración con otros sistemas: permitirá los sistema autónomo formen redes entre ellos para compartir información.

Figura 8.2.: Entrada al sistema

8.3. Seleccionar proyecto

Al dar click en “Seleccione un proyecto”, aparece la pantalla 8.3. Con esta pantalla bastará seleccionar el proyecto en el que se trabajará, y automáticamente el sistema lo convertirá en el proyecto activo, y ésto será reflejado en la interfaz como se ve en la pantalla 8.4.

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Seleccione un proyecto] Salir

Proyectos

Id	Nombre	Descripción
402	Proyecto 1	Descripción del proyecto 1

Insertar Editar Borrar Hacer activo Exportar XMLEx XMLIn Analizar

1 / 1 10 Rows

Figura 8.3.: Seleccionar proyecto

Al seleccionar el proyecto, también se desplegarán los datos esenciales de ese proyecto (pantalla 8.4 en la página siguiente).

The screenshot shows the main interface of the 'Sistema de Administración de Métricas para Equipos de Proyectos'. The user is logged in as 'Miguel Agustin Ruiz Velasco Sobrino' and is viewing the 'Proyecto 1' details. The interface includes a sidebar with navigation options like 'Proyectos', 'GQM', and 'Parámetros'. The main content area displays a table with one row for 'Proyecto 1' and a detailed view of its data, including description, date, and processes.

Id	Nombre	Descripcion
402	Proyecto 1	Descripción del proyecto 1

Datos del proyecto: Proyecto 1
 Descripción: Descripción del proyecto 1
 Fecha: 2009-02-17T15:48:59.133-06:00
Procesos
 Desarrollo: proceso de Desarrollo
 AdmProy: administración de proyectos

Figura 8.4.: Proyecto seleccionado

8.4. Metas

Al seleccionar la opción “Metas”, aparece la pantalla 8.5 mediante la cual, el usuario puede ver el detalle (8.6), agregar nueva (8.7), editar (8.8) o borrar las metas definidas en el sistema, como se ve en las pantallas .

The screenshot shows the 'Metas' screen within the same system. The user is still logged in as 'Miguel Agustin Ruiz Velasco Sobrino'. The main content area displays a table with one row for 'Meta uno' and a detailed view of its data, including description and ID.

Id	Nombre	Descripcion
402	Meta uno	descripcion de la meta uno uno

Figura 8.5.: Pantalla principal de metas

The screenshot shows the 'Detalle de una meta' page. At the top, the title 'Sistema de Administración de Métricas para Equipos de Proyectos' is displayed in a dark blue header. Below the header, the user's name 'Miguel Agustin Ruiz Velasco Sobrino' and the project name 'Proyecto 1' are shown, along with a 'Salir' button. On the left side, there are three main menu categories: 'Proyectos', 'GQM', and 'Parámetros', each with a list of sub-items. The main content area features a table titled 'Metas' with columns 'Id', 'Nombre', and 'Descripcion'. A single row is visible with '402', 'Meta uno', and 'descripcion de la meta uno uno'. Below the table is a toolbar with 'Insertar', 'Editar', and 'Borrar' buttons, and a pagination control showing '1 / 1' and '10 Rows'. The 'Datos de la meta: Meta uno' section contains the following text: 'Descripción: descripcion de la meta uno uno' and 'Fecha: 2009-02-17T15:43:00.804-06:00'.

Figura 8.6.: Detalle de una meta

The screenshot shows the 'Insertar meta' page. The header and user information are identical to Figure 8.6. The left sidebar menu is expanded to show 'Procesos', 'Tipos de Datos', 'Cuestionarios', and 'Calendarios' under the 'Parámetros' category, and 'Responder', 'Respuestas', 'Ver casos', and 'Sugerencias de técnicas de estimación' under the 'Registro' category. The main content area features the 'Metas' table with one row. Below the table is a toolbar with 'Insertar', 'Editar', and 'Borrar' buttons, and a pagination control showing '1 / 1' and '10 Rows'. The 'Datos de la meta' section contains a form with a 'Nombre:' label and an empty text input field, and a 'Descripcion:' label with a large empty text area. A 'Nuevo' button is located at the bottom right of the form.

Figura 8.7.: Insertar meta

The screenshot shows the main interface for managing questions. The header includes the system name and the user 'Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir'. On the left, there are navigation menus for 'Proyectos', 'GQM', and 'Parámetros'. The main content area displays a table of questions:

Id	Nombre	Descripcion
402	Pregunta dos	no hay pregunta uno! aja eh!

Below the table, there are controls for 'Insertar', 'Editar', and 'Borrar', along with pagination information: '1 / 1' and '10 Rows'. A detailed view for 'Pregunta dos' is shown below, with the description 'no hay pregunta uno! aja eh!' and a timestamp 'Fecha: 2009-02-17T15:43:45.654-06:00'.

Figura 8.9.: Pantalla principal de preguntas

The screenshot shows the main interface for managing goals. The header includes the system name and the user 'Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir'. On the left, there are navigation menus for 'Proyectos', 'GQM', and 'Parámetros'. The main content area displays a table of goals:

Id	Nombre	Descripcion
402	Meta uno	descripcion de la meta uno uno

Below the table, there are controls for 'Insertar', 'Editar', and 'Borrar', along with pagination information: '1 / 1' and '0 Rows'. A detailed view for 'Meta uno' is shown below, with the name 'Meta uno' and the description 'descripcion de la meta uno uno'. A 'Nuevo' button is visible at the bottom right of the form.

Figura 8.8.: Editar meta

8.5. Preguntas

Al seleccionar la opción “Preguntas” se pueden visualizar y editar las preguntas de alto nivel que se pretenden responder por medio del programa de métricas.

Aparece la pantalla 8.9 mediante la cual, el usuario puede ver el detalle (8.9), agregar nueva, editar (8.10) o borrar las metas definidas en el sistema, como se ve en las pantallas .

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas
- Ver casos
- Sugerencias de técnicas de estimación

Preguntas		
Id	Nombre	Descripcion
402	Pregunta dos	no hay pregunta uno!! aja eh!

Insertar Editar Borrar 1 / 1 10 0 Rows

Datos de la Pregunta

Nombre:

Descripcion:

Figura 8.10.: Editar pregunta

8.6. Métricas

Al seleccionar la opción “Métricas” se pueden visualizar y editar las mediciones que se tomarán y almacenaran.

Aparece la pantalla 8.11 mediante la cual, el usuario puede ver el detalle (8.11) al seleccionar una métrica, agregar una nueva métrica, editar una métrica (8.12) o borrar las metas definidas en el sistema, como se ve en las pantallas .

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos

Métricas		
Id	Nombre	Descripcion
402	Size	pregunta de la Medición de tamaño
402	Effort	te esforzaste?

Insertar Editar Borrar 1 / 2 10 0 Rows

Datos de la metrica: Size

Descripcion: pregunta de la Medición de tamaño
 Fecha: 2009-02-17T15:45:02.562-06:00 Explicación adicional: descripcion 2 de la medición de tamaño

Figura 8.11.: Pantalla principal de métricas

A diferencia de las anteriores, la primera descripción, es lo que el usuario verá en el cuestionario, mientras que la segunda es una explicación de la pregunta.

The screenshot shows the 'Sistema de Administración de Métricas para Equipos de Proyectos' interface. The main header displays the system name and the user 'Miguel Agustín Ruiz Velasco Sobrino' with a 'Salir' button. A sidebar on the left contains navigation menus for 'Proyectos', 'GQM', 'Parámetros', and 'Registro'. The main content area features a table of metrics and a detailed form for editing a metric.

Id	Nombre	Descripción
402	Effort	te esforzaste?
402	Defectos	Cuantos defectos tuvo el proyecto en el periodo
402	Size	pregunta de la Medición de tamaño

The 'Datos de la métrica' form includes the following fields:

- Nombre:** Size
- Texto de la pregunta:** Número de líneas de código programadas en el periodo
- Descripción:** descripcion 2 de la medición de tamaño

A 'Nuevo' button is located at the bottom right of the form.

Figura 8.12.: Editar métrica

8.7. Proyectos

Al seleccionar la opción “Proyectos” se pueden visualizar y editar las mediciones que se tomarán y almacenaran.

Aparece la pantalla 8.13 mediante la cual, el usuario puede ver los proyectos, el detalle (8.13) al seleccionar un proyecto, agregar un nuevo proyecto, editar un proyecto (8.15 en la página 89), borrarlo, exportar los datos a un DataWarehouse para su análisis (8.16 en la página 89), exportar los datos para almacenarlos offline o enviarlos como archivo xml (8.17 en la página 90), importar los datos de otro proyecto o invocar un asistente de análisis.

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

Seleccionar
Mis pendientes
Reportes

GQM

Metas
Preguntas
Métricas
Mapeo

Parámetros

Procesos
Tipos de Datos

Id	Nombre	Descripción
402	Proyecto 1	Descripción del proyecto 1

Insertar Editar Borrar Hacer activo Exportar XMLEx XMLIn Analizar 1 / 1 10 0 Rows

Datos del proyecto: Proyecto 1

Descripción: Descripción del proyecto 1
Fecha: 2009-02-17T15:48:59.133-06:00

Procesos

Desarrollo: proceso de Desarrollo
AdmProy: administración de proyectos

Figura 8.13.: Pantalla principal de proyectos

En la pantalla de insertar proyecto (8.14), se le tiene que dar un nombre y una descripción al proyecto, así como seleccionar los procesos que el proyecto tiene, entre todos los procesos que se han definido en el sistema.

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

Seleccionar
Mis pendientes
Reportes

GQM

Metas
Preguntas
Métricas
Mapeo

Parámetros

Procesos
Tipos de Datos
Cuestionarios
Calendarios

Registro

Responder
Respuestas
Ver casos
Sugerencias de técnicas de estimación

Id	Nombre	Descripción
402	Proyecto 1	Descripción del proyecto 1

Insertar Editar Borrar Hacer activo Exportar XMLEx XMLIn Analizar 1 / 1 10 0 Rows

Datos del Proyecto

Nombre:

Descripción:

Procesos: Desarrollo AdmProy

Nuevo

Figura 8.14.: Insertar proyecto

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

Seleccionar
Mis pendientes
Reportes

GQM

Metas
Preguntas
Métricas
Mapeo

Parámetros

Procesos
Tipos de Datos
Cuestionarios
Calendarios

Registro

Responder
Respuestas
Ver casos
Sugerencias de técnicas de estimación

Id	Nombre	Descripción
402	Proyecto 1	Descripción del proyecto 1

Insertar Editar Borrar Hacer activo Exportar XMLEx XMLIn Analizar 1/1 10 0 Rows

Datos del Proyecto

Nombre:

Descripción:

Procesos: Desarrollo AdmProy

Nuevo

Figura 8.15.: Editar proyecto

La opción “Exportar” (8.16) permite pasar los datos del proyecto seleccionado a un data warehouse, para su posterior análisis. Es necesario exportar los datos cada vez que cambien, porque no se actualizan automáticamente cuando cambian en el data warehouse.

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

Seleccionar
Mis pendientes
Reportes

GQM

Metas
Preguntas
Métricas
Mapeo

Parámetros

Procesos
Tipos de Datos

Id	Nombre	Descripción
402	Proyecto 1	Descripción del proyecto 1

Insertar Editar Borrar Hacer activo Exportar XMLEx XMLIn Analizar 1/1 10 0 Rows

Datos del proyecto: Proyecto 1

Descripción: Descripción del proyecto 1
Fecha: 2009-02-17T15:48:59.133-06:00

Procesos

Desarrollo: proceso de Desarrollo
AdmProy: administración de proyectos

Figura 8.16.: Exportar datos a un DataWarehouse

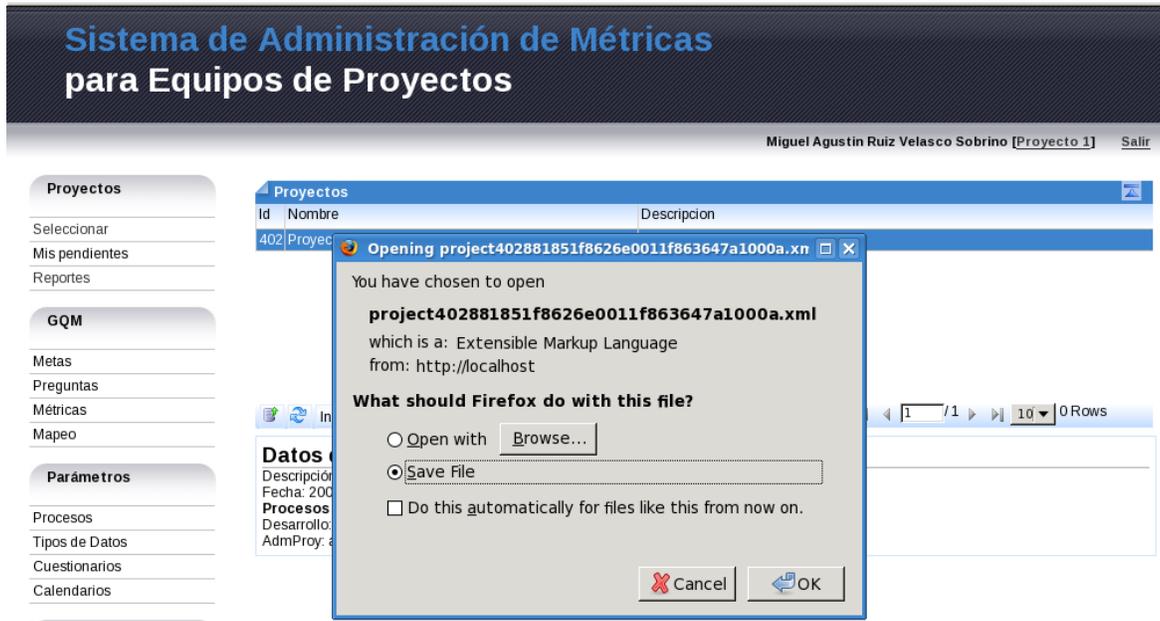


Figura 8.17.: Exportar proyecto

8.8. Cuestionarios

Al seleccionar la opción “Cuestionarios” se pueden visualizar y editar los cuestionarios que se tomarán y almacenarán.

Aparece la pantalla 8.18 mediante la cual, el usuario puede ver el detalle (8.18) al seleccionar un cuestionario, agregar un nuevo cuestionario (8.19 en la página siguiente), editar un cuestionario (8.20 en la página 92) o borrar un cuestionario definidos en el sistema, como se ve en las pantallas .

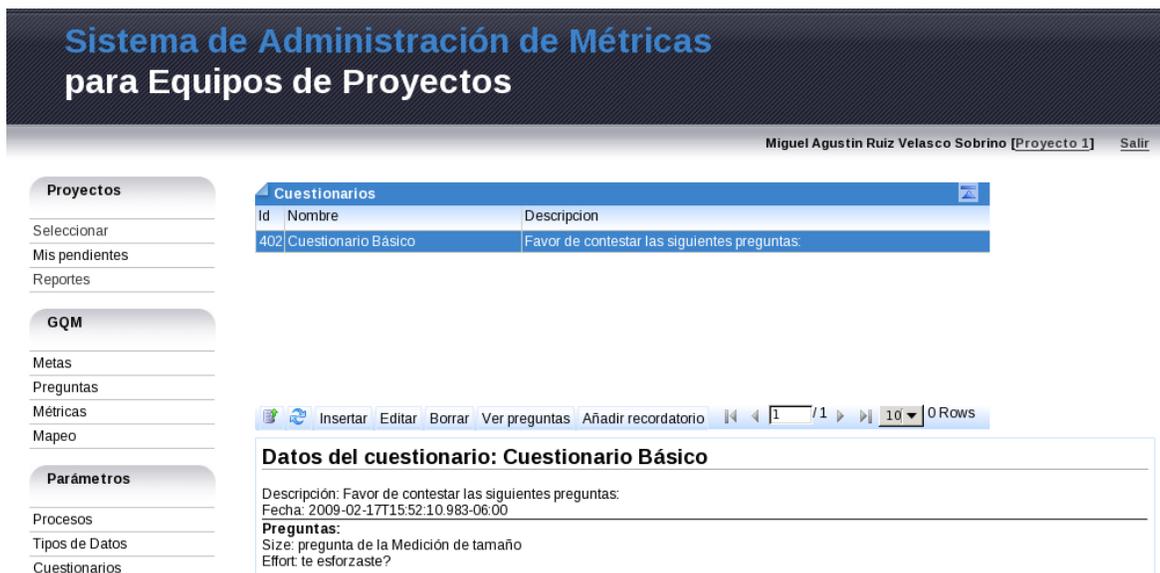


Figura 8.18.: Pantalla principal de cuestionarios

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas
- Ver casos
- Sugerencias de técnicas de estimación

Cuestionarios

Id	Nombre	Descripcion
402	Cuestionario Basico	Favor de contestar las siguientes preguntas:

Insertar Editar Borrar Ver preguntas Añadir recordatorio 1 / 1 10 0 Rows

Datos del Cuestionario

Nombre:

Descripcion:

Preguntas: Size Effort

Figura 8.19.: Insertar cuestionario

The screenshot displays the 'Sistema de Administración de Métricas para Equipos de Proyectos' interface. At the top, the title 'Sistema de Administración de Métricas para Equipos de Proyectos' is shown in a dark blue header. Below the header, the user's name 'Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1]' and a 'Salir' button are visible. The left sidebar contains several menu categories: 'Proyectos' (with sub-items: Seleccionar, Mis pendientes, Reportes), 'GQM' (with sub-items: Metas, Preguntas, Métricas, Mapeo), 'Parámetros' (with sub-items: Procesos, Tipos de Datos, Cuestionarios, Calendarios), and 'Registro' (with sub-items: Responder, Respuestas, Ver casos, Sugerencias de técnicas de estimación). The main content area shows a table titled 'Cuestionarios' with columns 'Id', 'Nombre', and 'Descripción'. The first row is highlighted in blue and contains the values '402', 'Cuestionario Básico', and 'Favor de contestar las siguientes preguntas:'. Below the table is a form titled 'Datos del Cuestionario' with the following fields: 'Nombre:' containing 'Cuestionario Básico', 'Descripción:' containing 'Favor de contestar las siguientes preguntas:', and 'Preguntas:' with checkboxes for 'Size' and 'Effort', both of which are checked. A 'Nuevo' button is located at the bottom right of the form. The top of the form area includes a toolbar with buttons for 'Insertar', 'Editar', 'Borrar', 'Ver preguntas', and 'Añadir recordatorio', along with a pagination control showing '1 / 1' and '10' rows.

Figura 8.20.: Editar cuestionario

8.9. Recordatorios

En la pestaña “Recordatorios”, se puede ver el status que guardan los distintos recordatorios que se han programado en el sistema, así como editarlos.

También se pueden programar recordatorios recurrentes directamente desde la pantalla de cuestionarios, con la ventaja que automáticamente se copian los datos del cuestionario para facilitar responderlo (8.21 en la página siguiente, 8.22 en la página 94, 8.23 en la página 95).

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

Seleccionar

Mis pendientes

Reportes

GQM

Metas

Preguntas

Métricas

Mapeo

Parámetros

Procesos

Tipos de Datos

Cuestionarios

Calendarios

Registro

Responder

Respuestas

Ver casos

Sugerencias de técnicas de estimación

Cuestionarios		
Id	Nombre	Descripcion
402	Cuestionario Básico	Favor de contestar las siguientes preguntas:

Insertar Editar Borrar Ver preguntas Añadir recordatorio

1 / 1 10 0 Rows

Datos del recordatorio

Nombre:

Receptores:

Tipo: Cuestionario

Id del cuestionario: 402881851f8626e0011f86398

Inicio:

Fin:

Tipo: Semanal

Proyecto:

Figura 8.21.: Programar una serie de recordatorios sobre un cuestionario

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] [Salir](#)

Proyectos

Seleccionar

Mis pendientes

Reportes

GQM

Metas

Preguntas

Métricas

Mapeo

Parámetros

Procesos

Tipos de Datos

Cuestionarios

Calendarios

Registro

Responder

Respuestas

Ver casos

Sugerencias de técnicas de estimación

Cuestionarios		
Id	Nombre	Descripcion
402	Cuestionario Básico	Favor de contestar las siguientes preguntas:

Insertar Editar Borrar Ver preguntas Añadir recordatorio

Datos del recordatorio

Nombre:

Receptores:

Tipo:

Id del cuestionario:

Inicio:

Fin:

Tipo:

Proyecto:

<<< marzo 2009 >>>

l	m	j	v	s	d
23	24	25	26	27	28
2	3	4	5	6	7
9	10	11	12	13	14
16	17	18	19	20	21
23	24	25	26	27	28

Nuevo

Figura 8.22.: Programar una serie de recordatorios sobre un cuestionario

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas
- Ver casos
- Sugerencias de técnicas de estimación

Cuestionarios

Id	Nombre	Descripción
402	Cuestionario Básico	Favor de contestar las siguientes preguntas:

Insertar Editar Borrar Ver preguntas Añadir recordatorio 1 / 1 10 0 Rows

Datos del recordatorio

Nombre: Recordatorio 1

Receptores: miguelrvs

Tipo: Cuestionario

Id del cuestionario: 402881851f8626e0011f86398

Inicio: 02/22/2009

Fin: 03/24/2009

Tipo: Semanal

Proyecto: Proyecto 1

Nuevo

Figura 8.23.: Programar una serie de recordatorios sobre un cuestionario

En la pestaña “Mis pendientes” se pueden ver los pendientes por reportar que tiene la persona. También se pueden ver el status que guarda cada una de los recordatorios (8.24).

Desde esa pantalla, se puede oprimir el botón “Responder” y automáticamente aparecerá el reporte solicitado listo para ser llenado (8.25).

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos

Pendientes

Id	Nombre	Status	Fecha
402	Recordatorio 1	started	2009-03-21T00:00:00

Responder 1 / 1 10 1 Rows

Datos del recordatorio: Recordatorio 1

Status: read

Fecha sugerida de realización: 2009-03-21T00:00:00-06:00

Fecha de realización:

Responder

Figura 8.24.: Listado de reportes pendientes de llenar

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas
- Ver reporte

Pendientes

Id	Nombre	Status	Fecha
402	Recordatorio 1	started	2009-03-21T00:00:00

Responder 1 / 1 10 1 Rows

Proyecto 1: Cuestionario Básico

Proceso: Desarrollo

Tipo de dato: Estimado

Fecha de reporte: 02/22/2009

pregunta de la Medición de tamaño:

te esforzaste?:

Salvar

Figura 8.25.: Responder un reporte pendiente directamente

8.10. Tipos

Al seleccionar la opción “Tipos” se pueden visualizar y editar los tipos de datos de datos que se utilizarán, como por ejemplo: meta, real, estimado, etc.

Aparece la pantalla 8.26 mediante la cual, el usuario puede ver el detalle (8.26) al seleccionar un tipo, agregar un nuevo tipo, editar un tipo (8.27) o borrar algún tipo definido en el sistema, como se ve en las pantallas .

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Tipos de Medición

Id	Nombre	Descripcion
402	Real	Dato real
402	Estimado	Dato estimado
402	Meta	Dato meta

Insertar Editar Borrar 1 / 3 10 0 Rows

Datos del tipo de medición: Real

Descripción: Dato real

Figura 8.26.: Pantalla principal de tipos

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas
- Ver casos
- Sugerencias de técnicas de estimación

Tipos de Medición

Id	Nombre	Descripción
402	Real	Dato real
402	Estimado	Dato estimado
402	Meta	Dato meta

Insertar Editar Borrar 1 / 3 10 0 Rows

Datos del tipo de dato

Nombre: Real

Dato real

Descripción:

Nuevo

Figura 8.27.: Editar tipo

8.11. Procesos

Al seleccionar la opción “Procesos” se pueden visualizar y editar los proceso o etapas que utiliza la empresa en sus actividades.

Aparece la pantalla 8.13 mediante la cual, el usuario puede ver el detalle (8.28) al seleccionar un proceso, agregar un nuevo proceso, editar un proceso (8.29) o borrar algún proceso definido en el sistema, como se ve en las pantallas .

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Procesos

Id	Nombre	Descripción
402	AdmProy	administración de proyectos
402	Desarrollo	proceso de Desarrollo

Insertar Editar Borrar 1 / 2 10 0 Rows

Datos del proceso: AdmProy

Descripción: administración de proyectos

Figura 8.28.: Pantalla principal de procesos

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

GQM

- Metas
- Preguntas
- Métricas
- Mapeo

Parámetros

- Procesos
- Tipos de Datos
- Cuestionarios
- Calendarios

Registro

- Responder
- Respuestas
- Ver casos
- Sugerencias de técnicas de estimación

Id	Nombre	Descripción
402	AdmProy	administración de proyectos
402	Desarrollo	proceso de Desarrollo

Insertar Editar Borrar 1 / 2 10 0 Rows

Datos del proceso

Nombre:

Descripción:

Figura 8.29.: Editar proceso

8.12. Respuestas

Por otra parte, también se pueden responder directamente cuestionarios en cualquier momento, no necesariamente usando un recordatorio. Para ello hay que dar click en “Responder”, y elegir el cuestionario deseado de una lista (8.30).

A continuación se muestra el cuestionario (8.31), en el que lo único adicional a las respuestas, es la fecha a la cual corresponden esas respuestas.

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustín Ruiz Velasco Sobrino [Proyecto 1] Salir

Proyectos

- Seleccionar
- Mis pendientes
- Reportes

Nombres Descripción

Cuestionario Básico Descripción: Favor de contestar las siguientes preguntas:

Figura 8.30.: Elegir cuestionario para responder

Sistema de Administración de Métricas para Equipos de Proyectos

Miguel Agustin Ruiz Velasco Sobrino [Proyecto 1] [Salir](#)

Proyectos

Seleccionar

Mis pendientes

Reportes

GQM

Metas

Preguntas

Métricas

Mapeo

Proyecto 1: Cuestionario Básico

Proceso:

Tipo de dato:

Fecha de reporte:

<< febrero 2009 >>

l	m	m	j	v	s	d
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1

pregunta de la Medición de tamaño:

te esforzaste?:

Figura 8.31.: Responder cuestionario

8.13. Análisis de datos gráfico

8.13.1. Entrada

Al abrir la página del sistema de inteligencia de negocios (BI- bussiness intelligence), nos aparece la pantalla de bienvenida del sistema (8.32), y nos pide nuestras credenciales para poder acceder al sistema (8.33). Una vez proporcionadas unas credenciales correctas, nos aparece el escritorio principal del sistema de BI (8.34).



Figura 8.32.: Portada del data-warehouse

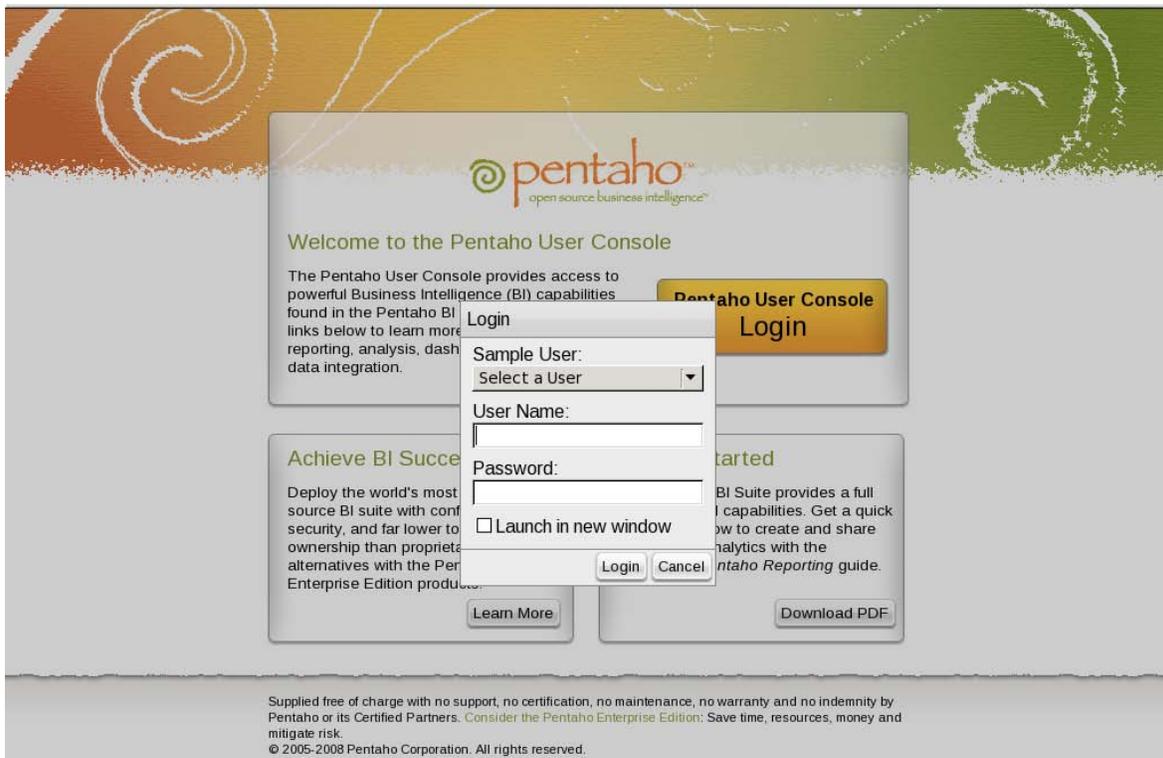


Figura 8.33.: Entrada al data-warehouse

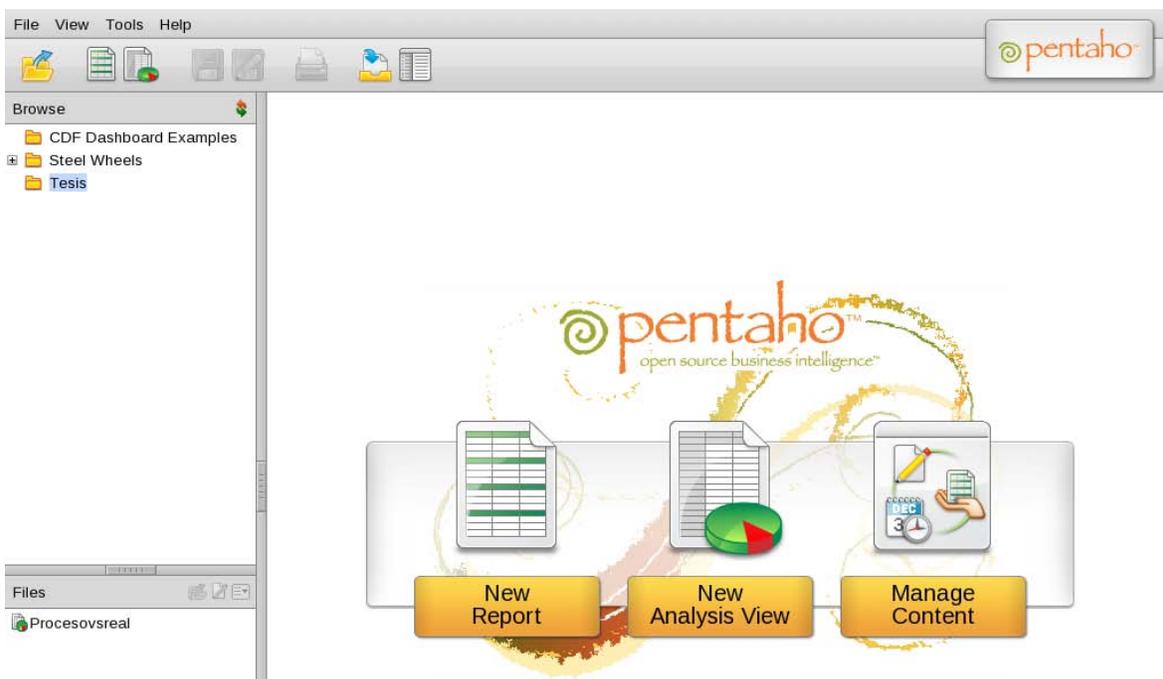


Figura 8.34.: Escritorio del data-warehouse

8.13.2. Generar un nuevo análisis OLAP

Para generar un nuevo análisis OLAP (On-line analysis processing) , se selecciona “New Analysis View” del escritorio (8.34) y nos aparece la pantalla para seleccionar el cubo básico que se personalizará durante el análisis (8.35).

El resultado de estas operaciones es el cubo básico que se muestra en la pantalla 8.36 que tiene como característica que todas las dimensiones están en las filas y “cerradas” (agrupadas).



Figura 8.35.: Interfaz para definir un nuevo análisis usando cubos OLAP

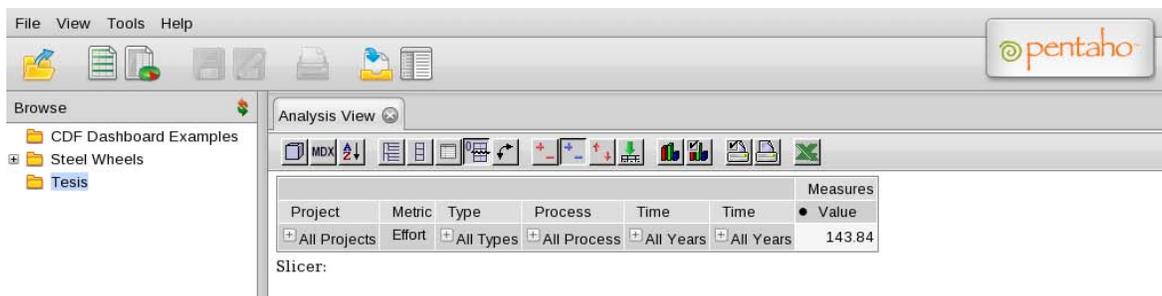


Figura 8.36.: Primer resultado del análisis

8.13.3. Reorganizar dimensiones

Una de las principales funcionalidades de OLAP son las tablas pivote, en donde se pueden cambiar los datos (dimensiones) entre columnas y filas, así como agruparlos y desagruparlos a diferentes niveles, y en las celdas resultantes aplicarles funciones matemáticas para hacer agregación, como sumas, promedios, desviaciones estándar, etc.

En este ejemplo vamos a comparar las dimensiones de tiempo de captura vs tiempo de origen de los datos.

Para ésto, hay que cambiar una de las dimensiones de tiempo a las columnas.

- El cubo comienza como se ve en la figura (8.37), con todas las dimensiones en las filas.
- Ahora hay que cambiar de forma que cada una de las dimensiones de tiempo quede en las filas y otra en las columnas 8.38 en la página siguiente.
- El resultado se puede apreciar en la figura 8.39.

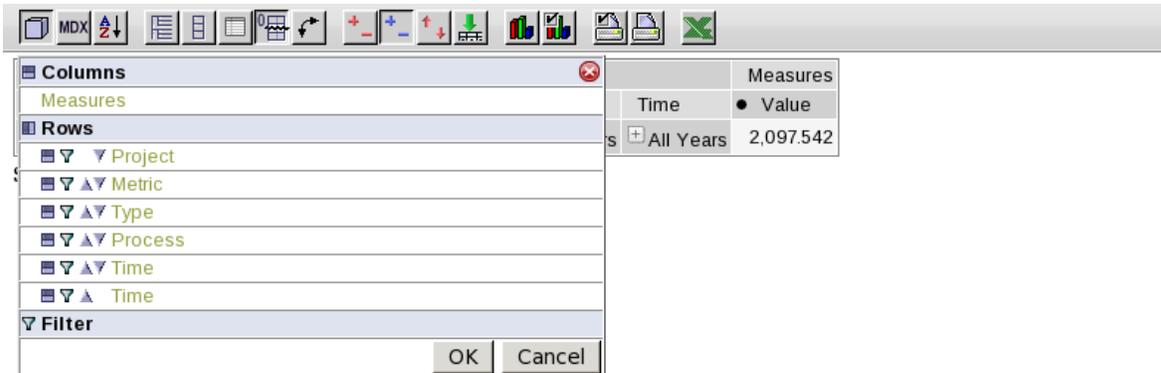


Figura 8.37.: Organización original de las dimensiones del cubo

Project	Metric	Type	Time	Process	Time				
					All Years	2009	1QTR	Jan	Feb
All Projects	All Metrics	All Types	All Years	All Process	2,097.542	2,076.542	2,076.542	1,845.2	231.342
				AdmProy	1,216	1,195	1,195	1,195	
				Desarrollo	881.542	881.542	881.542	650.2	231.342
			2009	All Process	2,097.542	2,076.542	2,076.542	1,845.2	231.342
				AdmProy	1,216	1,195	1,195	1,195	
				Desarrollo	881.542	881.542	881.542	650.2	231.342
			1QTR	All Process	2,097.542	2,076.542	2,076.542	1,845.2	231.342
				AdmProy	1,216	1,195	1,195	1,195	
				Desarrollo	881.542	881.542	881.542	650.2	231.342
			Feb	All Process	2,097.542	2,076.542	2,076.542	1,845.2	231.342
				AdmProy	1,216	1,195	1,195	1,195	
				Desarrollo	881.542	881.542	881.542	650.2	231.342

Figura 8.39.: Resultado de reorganizar las dimensiones del cubo

8.13.4. Generando una gráfica

La siguiente operación que se puede hacer es generar una gráfica para expresar de una forma más concisa y clara los datos contenidos en la tabla de datos.

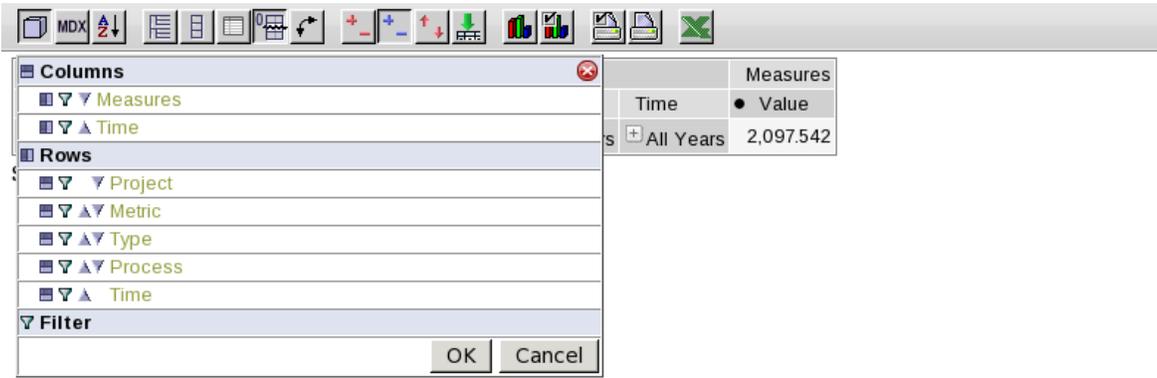


Figura 8.38.: Reorganizando las dimensiones del cubo

Se puede elegir el tipo de gráfica así como otros parámetros, en la pantalla 8.40, lo que produce la gráfica 8.41.

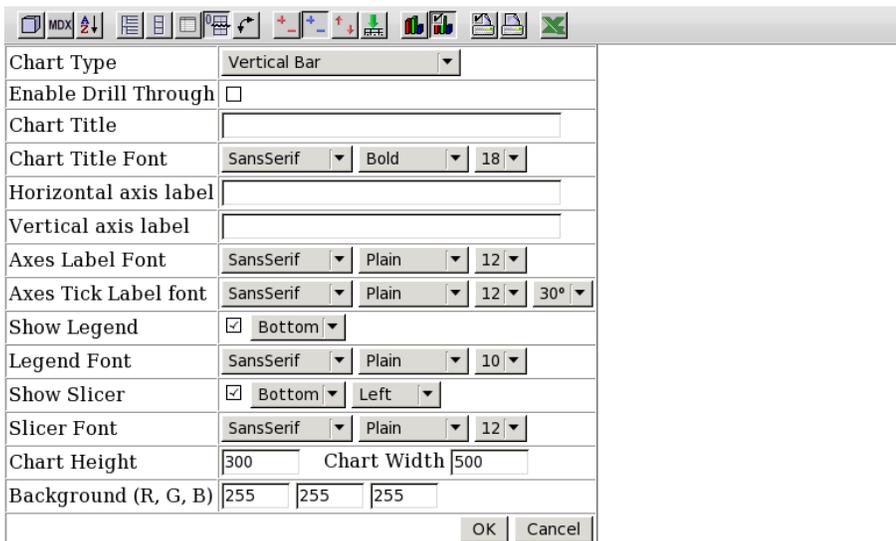


Figura 8.40.: Opciones de gráfica

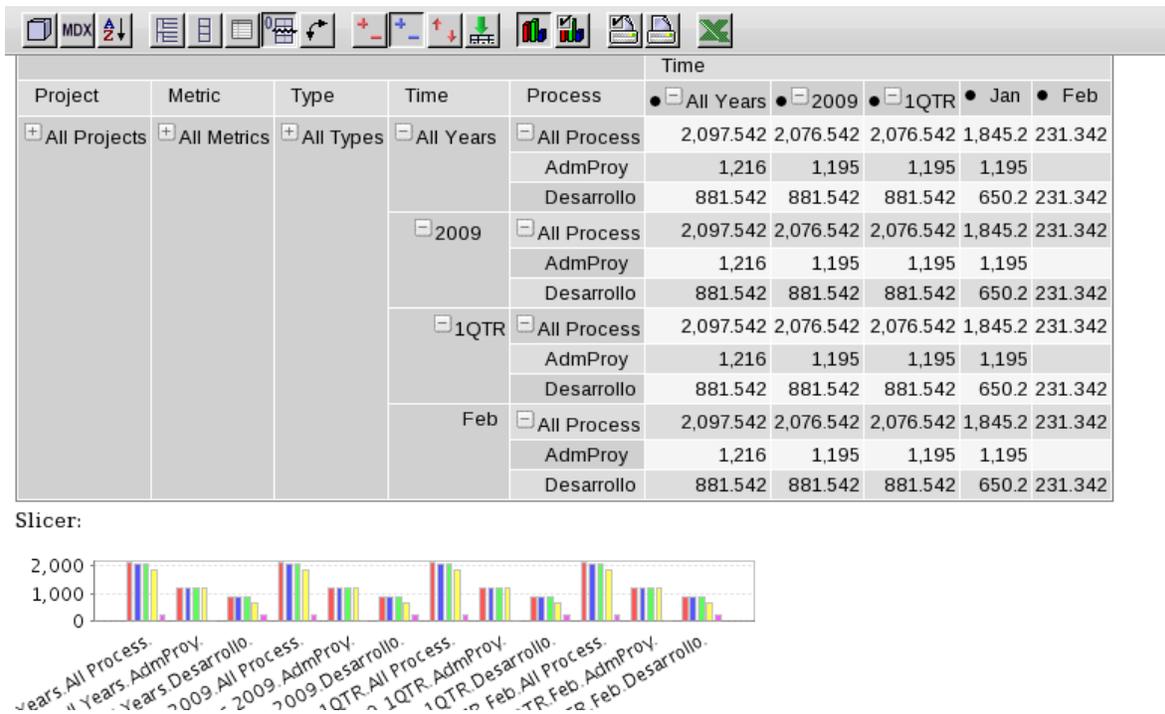


Figura 8.41.: Comparación entre tiempo capturado y tiempo del reporte por proceso

8.13.5. Guardar el análisis

Finalmente se puede salvar el análisis para su uso futuro, asignándole un nombre (8.42).

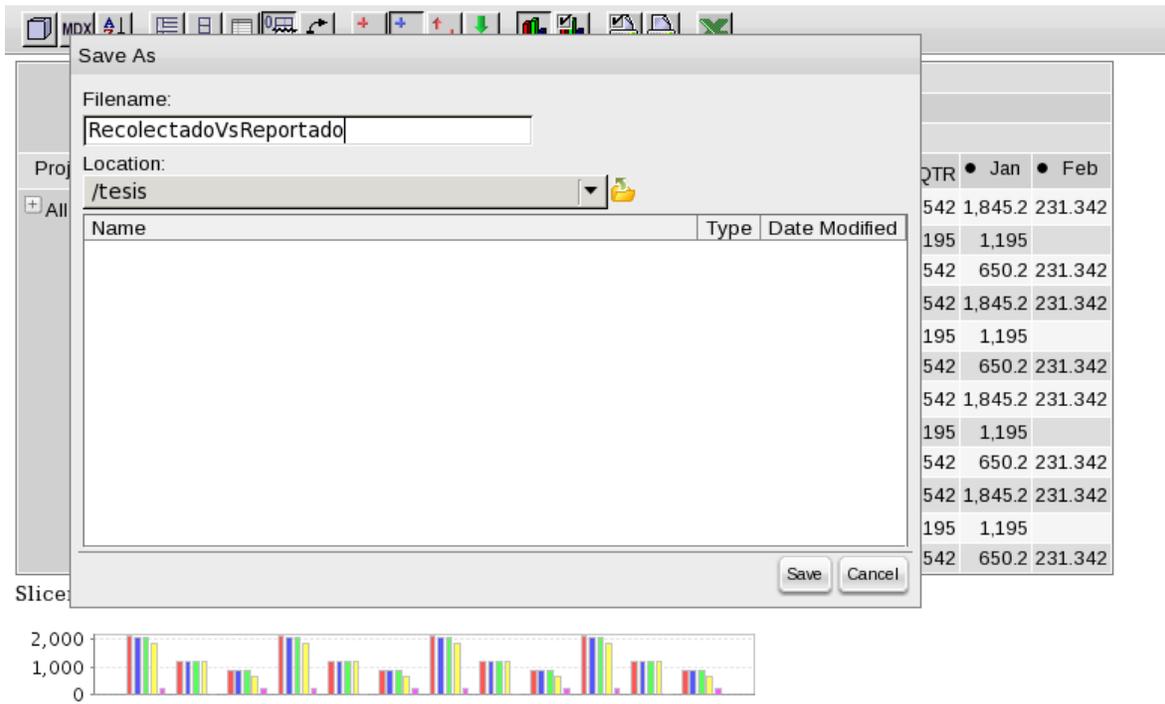


Figura 8.42.: Guardar el cubo para volverlo a usar

8. *Manual de usuario*

Habiendo quedado ya todo planteado, sólo resta enunciar las conclusiones, lo cual se hará a continuación.

Conclusiones

Las mediciones pueden ser una herramienta muy útil de la administración de proyectos si son recolectadas y analizadas de forma correcta.

En la opinión de M. en C. Cecilia Perez Colín, que trabaja en el departamento de medición en IBM y que colaboró en el planteamiento de los objetivos de la presente. La herramienta realizada junto con la tesis, y de la que forma parte integral, cumple con los objetivos que le fueron planteados originalmente, permitiendo simplificar los procesos de adquisición, almacenaje y análisis de los datos derivados de las mediciones. No hemos encontrado otros sistemas (propietarios o gratuitos) que realicen este conjunto de funciones, sobre todo en un entorno web.

En nuestra búsqueda tampoco hemos encontrado un sistema que permita accederlo desde una plataforma que no sea una PC, dándole al sistema una gran ventaja hacia el futuro en que las aplicaciones no necesariamente se accederán desde una PC de escritorio o laptop, sino que se usarán directamente desde dispositivos móviles desde cualquier sitio.

Para la alta dirección, la utilidad de las mediciones se incrementa conforme disminuye el contacto cotidianamente cara a cara entre sus integrantes, ya sea por su distribución geográfica, por el aumento en la cantidad de sus miembros, o por otros factores, como diferentes horarios laborales.

Para los estimadores de proyectos, el tener un mayor acceso a datos relevantes y actuales del desempeño de la organización incrementa la certidumbre y exactitud de sus modelos de estimación.

A las áreas de evaluación de desempeño les permite uniformizar los datos que les son reportados de los distintos proyectos, validarlos, evaluar su calidad, etc; para después proporcionar esos datos a las demás áreas de la organización.

Para los líderes de proyecto el tener herramientas para analizar el rendimiento de su propio proyecto, y poderlo comparar con otros proyectos presentes y pasados permite que se forme un criterio acerca de su propio proyecto.

Al equipo de proyecto le permite comprender el origen de ciertas decisiones que se toman, así como explorar alternativas para proponerlas a los líderes de proyecto y a niveles superiores de la jerarquía organizacional.

Por lo tanto una herramienta de este tipo puede reportar beneficios a todos los niveles y a todos los roles dentro de una organización.

El uso de un diseño orientado a objetos y en capas permite realizar cambios menores a la aplicación sin afectar otras secciones de la aplicación.

El uso de un diseño modular, en donde hay una separación clara entre los distintos módulos, permite una reutilización más fácil, incluso sin tener que recompilar la aplicación.

Los aspectos permiten separar ciertas secciones (“cross cutting concerns”) de la aplicación que provocarían que códigos destinados a diferentes funcionalidades quedaran mezclados (“tangled”).

El hecho de que la aplicación esté diseñada para ser integrada a otros sistemas, tanto para tomar datos de ellos, como para alimentarlos, permite romper con el aislamiento típico de los sistemas que ayudan a la toma de mediciones.

Derivado de esto, la capacidad de tomar datos de cualquier sistema que los pueda exportar, convierte a que los mismos sistemas que ya se usan en los procesos de proyecto en valiosas herramientas de recolección de datos.

Por otro lado, la capacidad de exportar los datos en los formatos nativos de otros sistemas, posibilita el uso de sistemas externos, cuya funcionalidad sería muy costosa de reproducir como parte integral

del sistema, ayudando a que los usuarios puedan usar las herramientas que mejor conozcan y a mejorar las capacidades y la utilidad con un esfuerzo relativamente bajo.

En cuanto al método originalmente planteado como uno de los objetivos de esta tesis, el cual ayudaría “a seleccionar métricas, tomar y analizar medidas de sus procesos y productos, y convertirlas en información que facilite la estimación de proyectos, procesos y productos futuros”; hice un descubrimiento sorprendente: que no puede haber tal. La diversidad de objetivos, necesidades de información, niveles de conocimiento y madurez en el tema de las mediciones, y una infinidad de factores (a diferentes niveles grupales e individual) hacen que no se pueda escribir una receta del tipo “one size fits all”. Lo que sí es universal es apoyar la formación de competencias (capacidad de saber hacer algo con calidad) en las personas que realizarán actividades relacionadas con mediciones, de forma que los integrantes del equipo puedan elevar su nivel entre sí y a los nuevos miembros generando y difundiendo conocimientos compartidos.

Trabajo futuro

Conocer qué otros tipos de análisis y modelos se usan en la vida real, ya sea de forma automatizada o manual, para adicionar la herramienta con módulos adecuados para realizar estos análisis.

Implantar la herramienta en una empresa real, ver su utilidad, los problemas que genera su uso, y aprovechar la experiencia ganada para mejorar la herramienta.

Investigar las necesidades de mediciones de otros sectores, como por ejemplo empresas que implementen ISO 9000, con el fin de verificar si la herramienta puede tener aplicación más allá de la medición en software.

Bibliografía

- [aja] Ajax. Available from: [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)).
- [aop] Aspect-oriented programming. Available from: http://en.wikipedia.org/wiki/Aspect-oriented_programming.
- [aspa] Aspectj. Available from: <http://en.wikipedia.org/wiki/Aspectj>.
- [aspb] The aspectj programming guide. Available from: <http://www.eclipse.org/aspectj/doc/released/progguide/starting-aspectj.html>.
- [Ber02] Ralph Bergmann. *Experience Management, foundations, developement, methodology and internet based applications*. Number 2432 in LNAI. Springer Berlag, 2002.
- [BK07] Christian Bauer and Gavin King. *Java Persistence with Hibernate*. Manning, 2nd edition, 2007.
- [clo] Cloud computing. Available from: http://en.wikipedia.org/wiki/Cloud_computing.
- [dem] W. edwards deming. Available from: http://en.wikipedia.org/wiki/W._Edwards_Deming#Deming.27s_14_Points.
- [dro] Drools. Available from: <http://en.wikipedia.org/wiki/Drools>.
- [for] Forward chaining. Available from: http://en.wikipedia.org/wiki/Forward_chaining.
- [Fow04] Martin Fowler. Inversion of control containers and the dependency injection pattern. 2004. Available from: <http://www.martinfowler.com/articles/injection.html>.
- [FP97] Norman Fenton and Shari Pfleeger. *Software Metrics: A rigorous and practical approach*. PWS Publishing Company, 2nd edition, 1997.
- [Gar05] Jesse James Garrett. Ajax: A new approach to web applications. 2005. Available from: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>.
- [GR95] Adele Goldberg and Kenneth S. Rubin. *Succeeding with objects, decision frameworks for project management*. Addison Wesley, 1995.
- [hib] Hibernate. Available from: [http://en.wikipedia.org/wiki/Hibernate_\(Java\)](http://en.wikipedia.org/wiki/Hibernate_(Java)).
- [Joh03] Rod Johnson. Introduction to the spring framework. 2003. Available from: <http://www.theserverside.com/tt/articles/article.tss?l=SpringFramework>.
- [KM90] Michel Klein and Leif B Methlie. *Expert Systems: a decision support approach*. Addison Wesley, 1990.
- [Leb07] Stanley Lebar. The saga of the lost space tapes. 2007. Available from: <http://www.washingtonpost.com/wp-dyn/content/article/2007/01/30/AR2007013002065.html>.
- [McC06] Steve McConnell. *Software Estimation*. Microsof press, 2006.

Bibliografía

- [McG85] Douglas McGregor. *The Human Side of Enterprise: 25th Anniversary Printing*. McGraw-Hill/Irwin, 1st edition, 1985.
- [MCJ⁺02] John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, and Fred Hall. *Practical software Measurement*. Addison Wesley, 2002.
- [MSK02] Dave Marcus, Steve Smith, and Mahan Khalsa. *Business Think*. Flanklink Covey, 2002.
- [Ric06] Chris Richardson. *POJOs in Action*. Manning, 2006.
- [Ros09] Jonathan Rosenberg. From the height of this place. 2009. Available from: <http://googleblog.blogspot.com/2009/02/from-height-of-this-place.html>.
- [Ser03] Roberto Servitje. *Bimbo: Estrategia de Éxito Empresarial*. Prentice Hall, 2003.
- [Ser08] Marcellin Jacques Sergio. *Notas del Curso: Construcción de Sistemas Expertos*. Posgrado en Ciencia e Ingeniería de la Computación, UNAM, 2008.
- [Spr08] SpringSource. About spring. 2008. Available from: <http://www.springsource.org/about>.
- [str] Struts 2. Available from: <http://en.wikipedia.org/wiki/Struts2>.
- [teo] Theory z. Available from: http://en.wikipedia.org/wiki/Theory_Z.
- [Wal08] Craig Walls. *Spring in Action*. Manning, 2nd edition, 2008.
- [Wil97] Adrian Wilkinson. *Empowerment: theory and practice*. 1997.

A. Casos de Uso

A.1. Caso de uso general

El caso de uso general detalla las distintas áreas funcionales en las que está distribuido el sistema y que se explicarán más adelante.

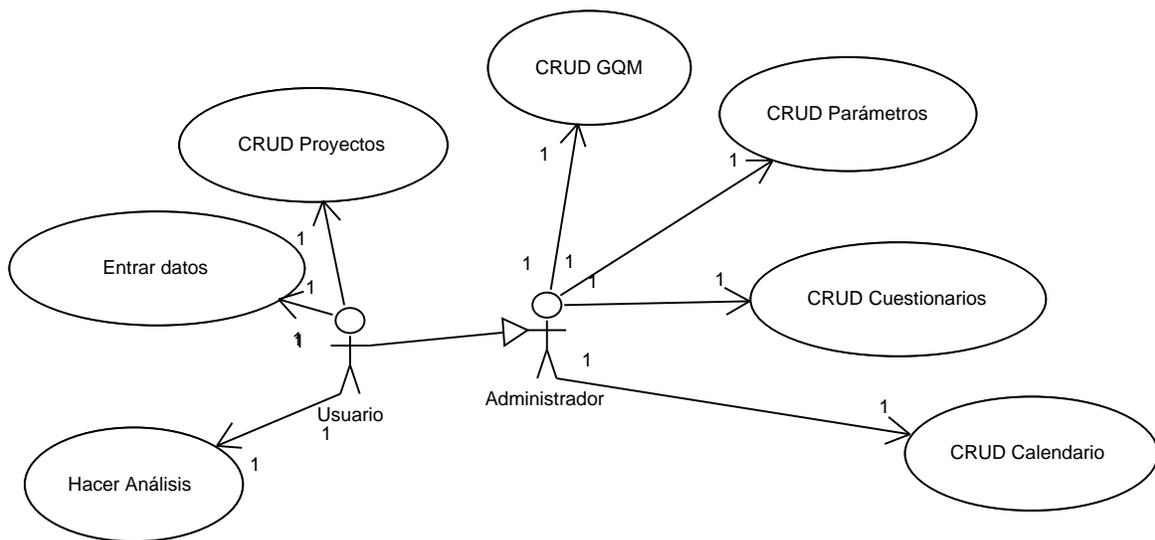


Figura A.1.: Diagrama general de Casos de uso

A.2. Caso de uso CRUD GQM

Este caso de uso tiene como único propósito agrupar de forma lógica otros casos de uso que serán descritos más adelante.

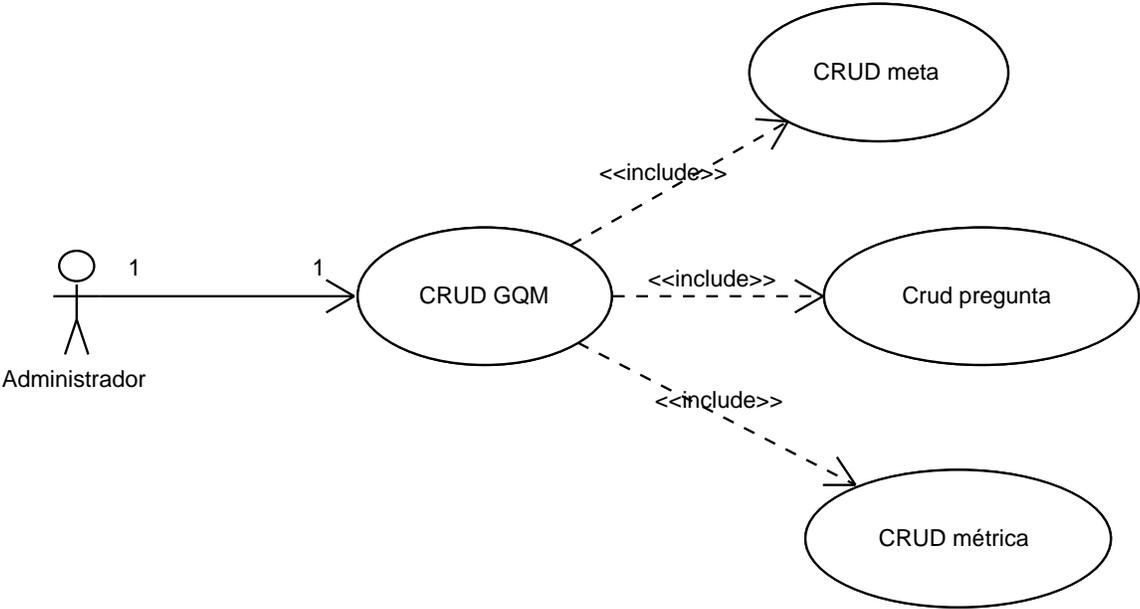


Figura A.2.: Caso de uso GQM

A.2.1. CRUD Meta

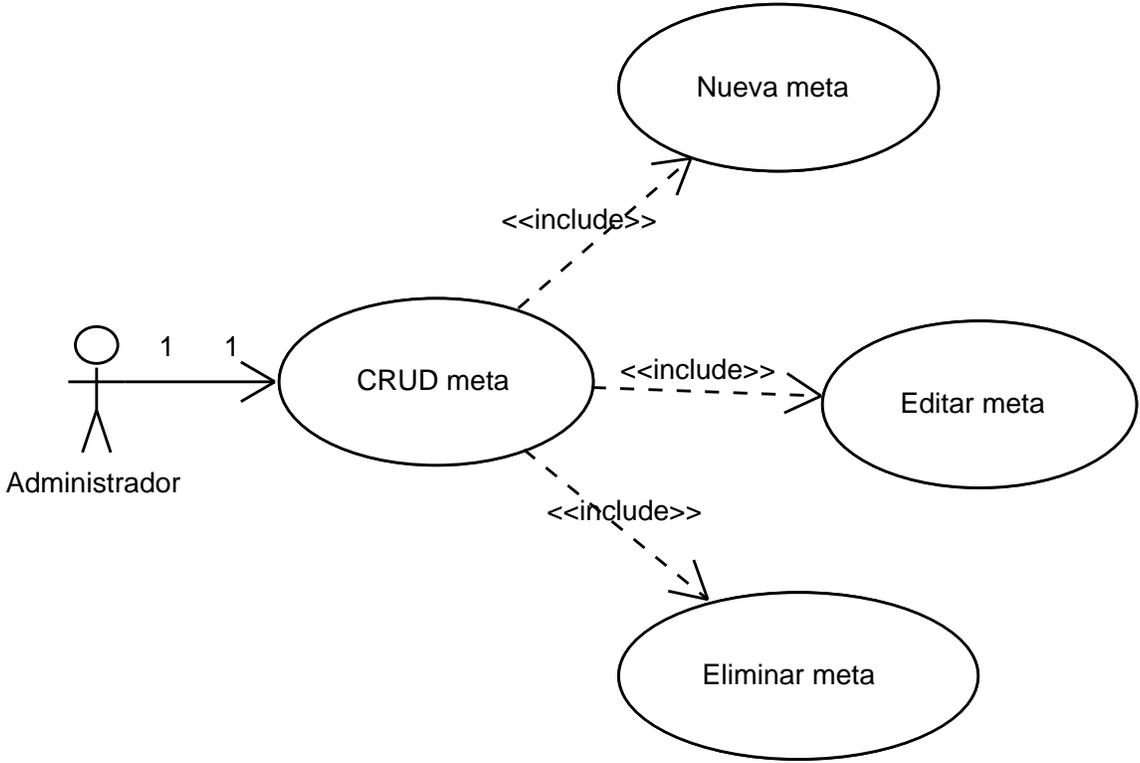


Figura A.3.: Caso de uso: CRUD meta

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador puede añadir, modificar y editar las metas dentro del sistema.

Precondiciones El administrador se encuentra autenticado como tal.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Metas”.
2. El sistema le muestra al usuario un menú con las metas que tiene registradas el sistema, y un conjunto de acciones que puede realizar con esas metas.
3. Si el usuario desea agregar una meta, el caso de uso se puede detener aquí.
4. El usuario selecciona una meta.
5. El sistema le despliega una panel con el detalle de la meta al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Postcondición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.1.1. Nueva Meta

Diagrama Véase diagrama A.3 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea añadir una meta nueva dentro del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD metas”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir meta”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda la nueva meta y actualiza el listado de metas. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.1.2. Editar Meta

Diagrama Véase diagrama A.3 en la página 112

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar una meta existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD metas”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una meta y elige la opción “Editar meta”. Si el usuario no selecciona una meta, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla con los datos de la meta que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda la meta editada y actualiza el listado de metas. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.1.3. Borrar Meta

Diagrama Véase diagrama A.3 en la página 112

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar una meta existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD metas”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una meta y elige la opción “Borrar meta”. Si el usuario no selecciona una meta, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra la meta del sistema y actualiza el listado de metas.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.2. CRUD Pregunta

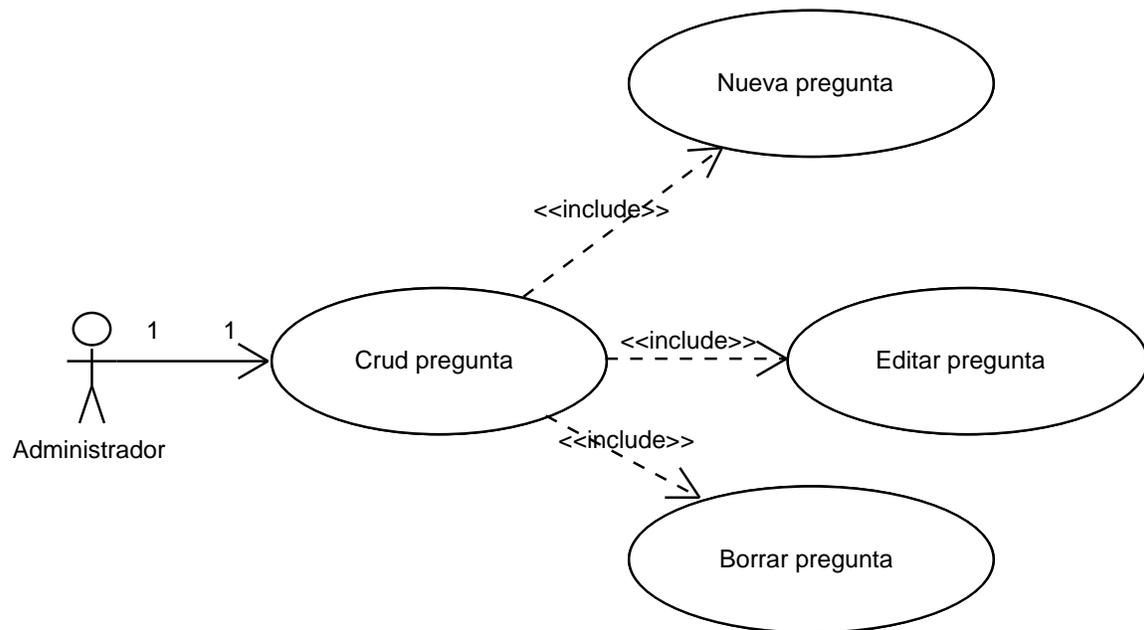


Figura A.4.: Caso de uso: CRUD pregunta

Diagrama**Actor primario** Administrador**Actor secundario** Ninguno**Descripción** El administrador puede añadir, modificar y editar las preguntas dentro del sistema.**Precondiciones** El administrador se encuentra autenticado como tal.**Flujo de eventos normales y excepcionales**

1. El usuario elige la opción "Preguntas".
2. El sistema le muestra al usuario un menú con las preguntas que tiene registradas el sistema, y un conjunto de acciones que puede realizar con esas metas.
3. Si el usuario desea agregar una pregunta, el caso de uso se puede detener aquí.
4. El usuario selecciona una pregunta.
5. El sistema le despliega una panel con el detalle de la pregunta al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Postcondición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.2.1. Nueva Pregunta

Diagrama Véase diagrama A.4 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea añadir una pregunta nueva dentro del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD pregunta”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir pregunta”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda la nueva pregunta y actualiza el listado de metas. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.2.2. Editar Pregunta

Diagrama Véase diagrama A.4 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar una pregunta existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD preguntas”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una pregunta y elige la opción “pregunta meta”. Si el usuario no selecciona una pregunta, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla con los datos de la meta que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda la pregunta editada y actualiza el listado de preguntas. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.2.3. Borrar Pregunta

Diagrama Véase diagrama A.4 en la página 115

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar una pregunta existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD preguntas”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una pregunta y elige la opción “Borrar pregunta”. Si el usuario no selecciona una pregunta, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra la pregunta del sistema y actualiza el listado de pregunta.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

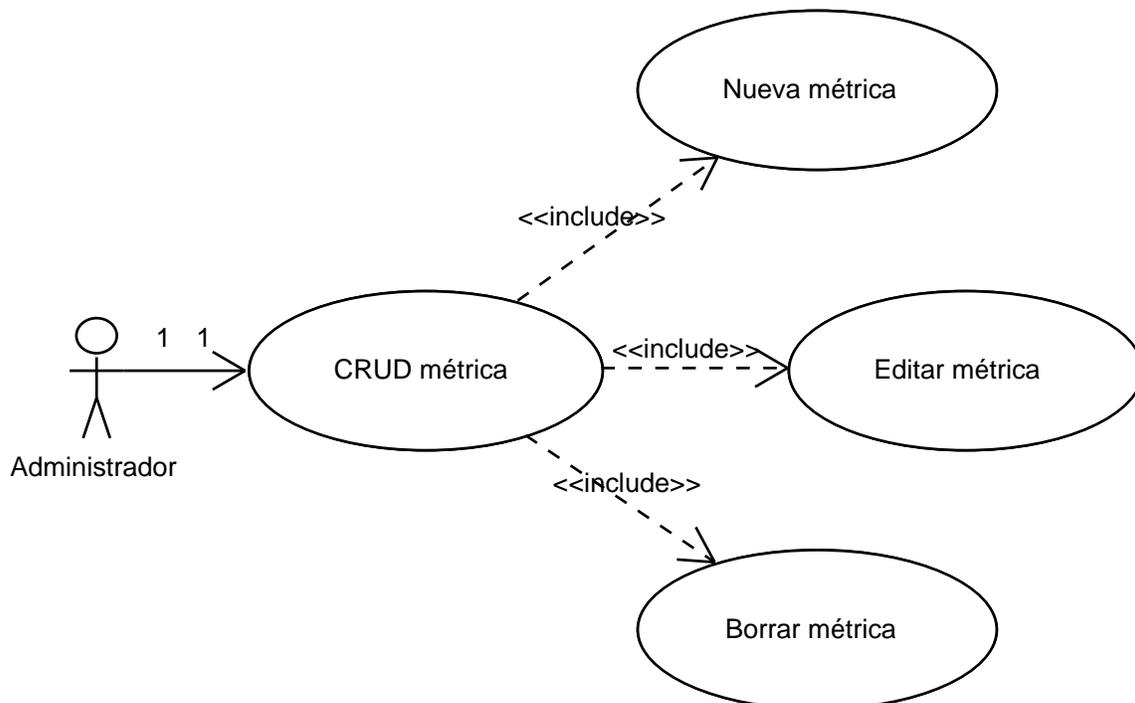
A.2.3. CRUD Métrica

Figura A.5.: Caso de uso CRUD métricas

Diagrama

A. Casos de Uso

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador puede añadir, modificar y editar las métricas dentro del sistema.

Precondiciones El administrador se encuentra autenticado como tal.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Métrica”.
2. El sistema le muestra al usuario un menú con las métricas que tiene registradas el sistema, y un conjunto de acciones que puede realizar con esas métricas.
3. Si el usuario desea agregar una métrica, el caso de uso se puede detener aquí.
4. El usuario selecciona una métrica.
5. El sistema le despliega una panel con el detalle de la métrica al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Postcondición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.3.1. Nueva Métrica

Diagrama Véase diagrama A.5 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea añadir una métrica nueva dentro del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD métricas”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir métrica”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda la nueva métrica y actualiza el listado de métricas. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.3.2. Editar Métrica

Diagrama Véase diagrama A.5 en la página 117

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar una métrica existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD métricas”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una métrica y elige la opción “Editar métrica”. Si el usuario no selecciona una métrica, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla con los datos de la métrica que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda la métrica editada y actualiza el listado de métricas. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.2.3.3. Borrar Métrica

Diagrama Véase diagrama A.5 en la página 117

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar una métrica existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD métricas”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una métrica y elige la opción “Borrar métrica”. Si el usuario no selecciona una métrica, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra la métrica del sistema y actualiza el listado de métricas.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3. Caso de uso Parámetros

Este caso de uso tiene como único propósito agrupar de forma lógica otros casos de uso que serán descritos más adelante.

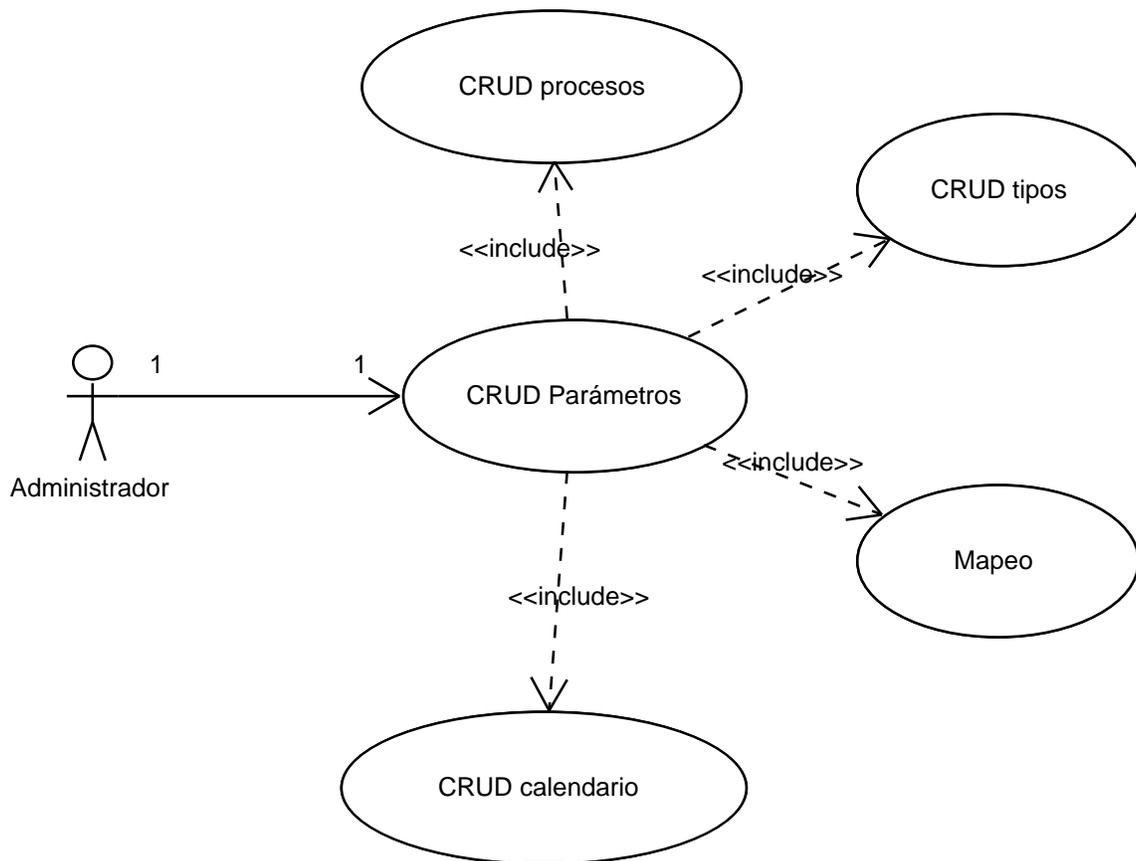


Figura A.6.: Caso de uso de parámetros del sistema

A.3.1. CRUD Procesos

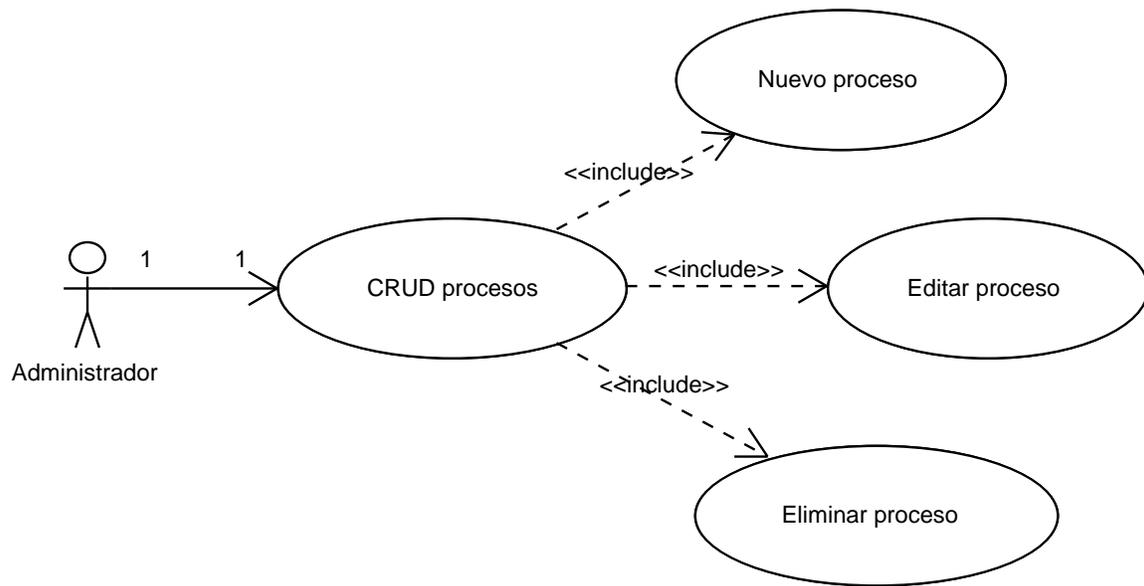


Figura A.7.: Caso de uso CRUD procesos

Diagrama**Actor primario** Administrador**Actor secundario** Ninguno**Descripción** El administrador puede añadir, modificar y editar los procesos dentro del sistema.**Precondiciones** El administrador se encuentra autenticado como tal.**Flujo de eventos normales y excepcionales**

1. El usuario elige la opción "Procesos".
2. El sistema le muestra al usuario un menú con los procesos que tiene registradas el sistema, y un conjunto de acciones que puede realizar con estos procesos.
3. Si el usuario desea agregar un proceso, el caso de uso se puede detener aquí.
4. El usuario selecciona un proceso.
5. El sistema le despliega una panel con el detalle del proceso al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Postcondición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.1.1. Nuevo Proceso

Diagrama Véase diagrama A.7 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea añadir un proceso nuevo dentro del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Procesos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir proceso”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda el nuevo proceso y actualiza el listado de procesos. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.1.2. Editar Proceso

Diagrama Véase diagrama A.7 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar un proceso existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Procesos”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una métrica y elige la opción “Editar proceso”. Si el usuario no selecciona un proceso, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla con los datos del proceso que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda el proceso editado y actualiza el listado de procesos. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.1.3. Borrar Proceso

Diagrama Véase diagrama A.7 en la página 121

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar un proceso existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Proceso”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un proceso y elige la opción “Borrar proceso”. Si el usuario no selecciona un proceso, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra el proceso del sistema y actualiza el listado de procesos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.2. CRUD Tipos

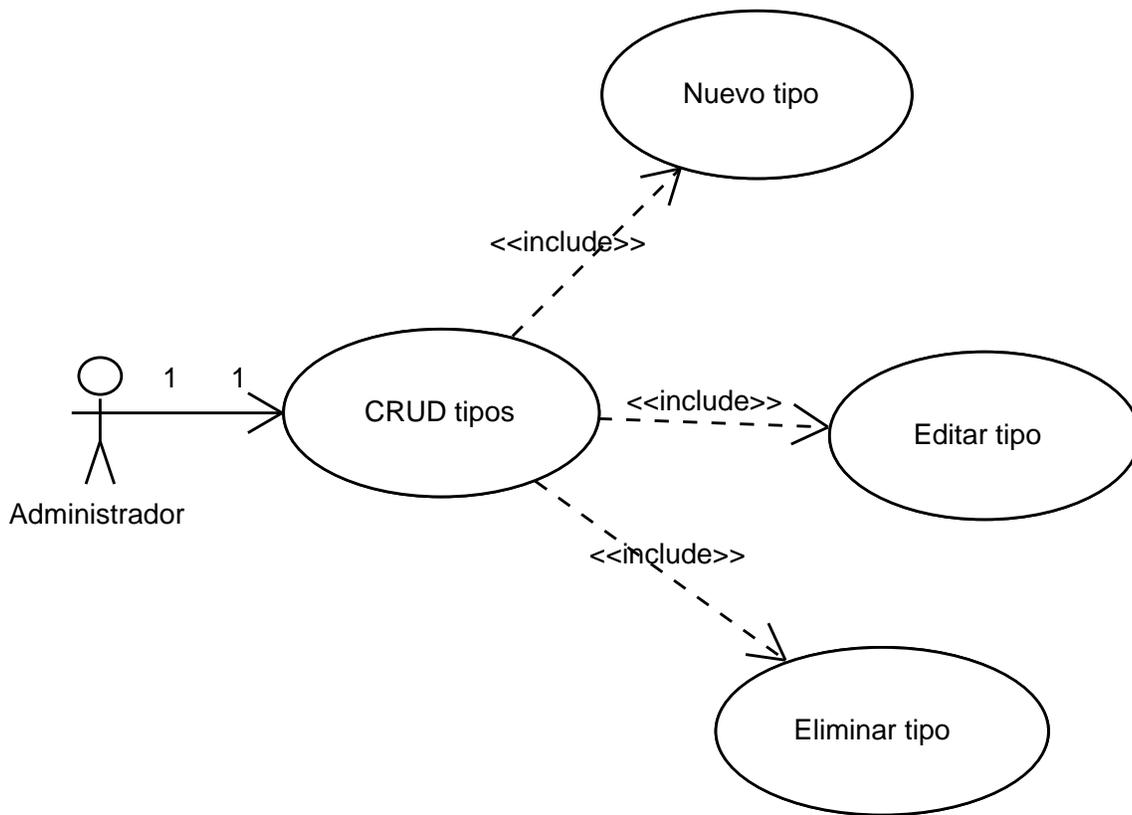


Figura A.8.: Caso de uso de tipos

Diagrama

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador puede añadir, modificar y editar los tipos dentro del sistema.

Precondiciones El administrador se encuentra autenticado como tal.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción "Tipos".
2. El sistema le muestra al usuario un menú con los tipos que tiene registradas el sistema, y un conjunto de acciones que puede realizar con estos tipos.
3. Si el usuario desea agregar un tipo, el caso de uso se puede detener aquí.
4. El usuario selecciona un tipo.
5. El sistema le despliega una panel con el detalle del proceso al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Postcondición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.2.1. Nuevo Tipo

Diagrama Véase diagrama A.8 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea añadir un tipo nuevo dentro del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Tipos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir tipo”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda el nuevo tipo y actualiza el listado de tipo. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.2.2. Editar Tipo

Diagrama Véase diagrama A.8 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar un tipo existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Tipos”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un tipo y elige la opción “Editar tipo”. Si el usuario no selecciona un tipo, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla con los datos del tipo que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda el tipo editado y actualiza el listado de tipos. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.2.3. Borrar Tipo

Diagrama Véase diagrama A.8 en la página 124

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar un tipo existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD tipo”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una tipo y elige la opción “Borrar tipo”. Si el usuario no selecciona un tipo, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra el tipo del sistema y actualiza el listado de tipos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.3. Calendario

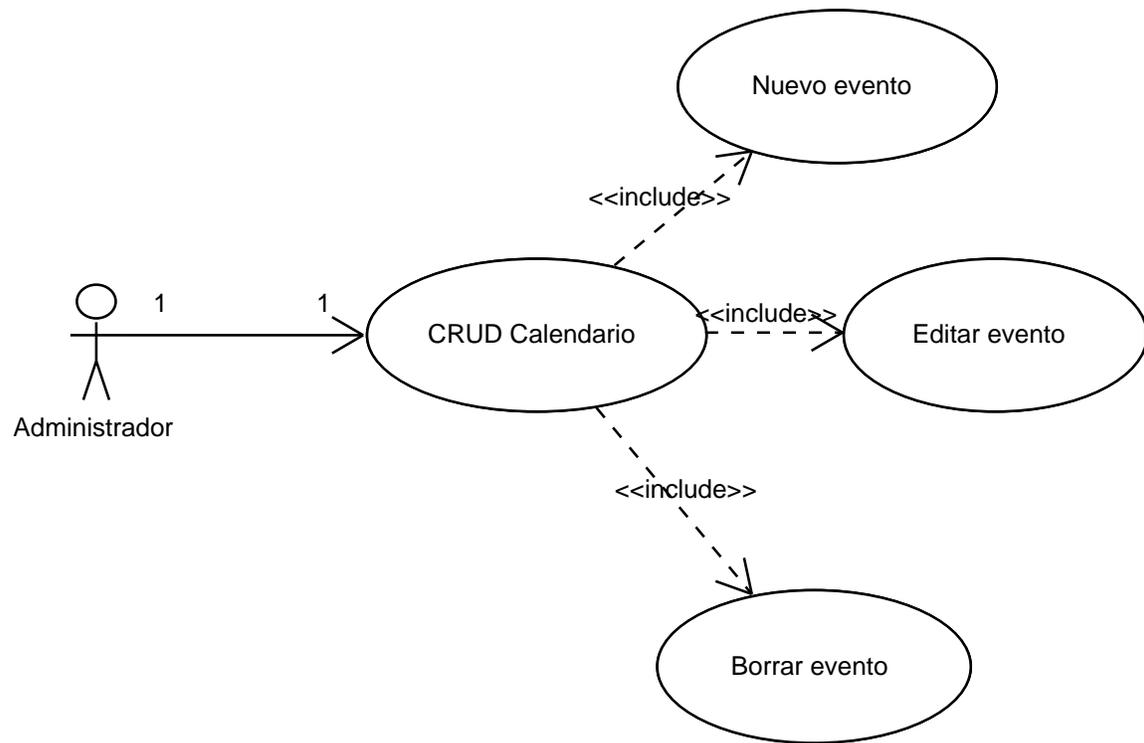


Figura A.9.: Caso de uso de calendario

Diagrama**Actor primario** Administrador**Actor secundario** Ninguno**Descripción** El administrador puede añadir, modificar y editar los eventos del calendario de un proyecto.**Precondiciones** El administrador se encuentra autenticado como tal.
El administrador tiene activo (seleccionado) un proyecto.**Flujo de eventos normales y excepcionales**

1. El usuario elige la opción "Calendario".
2. El sistema le muestra al usuario un menú con los eventos del calendario que tiene registradas el sistema, y un conjunto de acciones que puede realizar con estos procesos.
3. Si el usuario desea agregar un evento al calendario, el caso de uso se puede detener aquí.
4. El usuario selecciona un evento del calendario.
5. El sistema le despliega una panel con el detalle del eventos del calendario al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Postcondición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.3.1. Nuevo Evento del Calendario

Diagrama Véase diagrama A.9 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea añadir un evento nuevo al calendario de un proyecto.

Precondiciones El administrador terminó el caso de uso “Calendario”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda el nuevo evento del calendario y actualiza el listado de evento del calendario. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.3.2. Editar Evento del Calendario

Diagrama Véase diagrama A.9 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar un evento del calendario existente de un proyecto.

Precondiciones El administrador terminó el caso de uso “Calendario”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona una evento del calendario y elige la opción “Editar ”. Si el usuario no selecciona un evento del calendario, el sistema le avisa que debe seleccionar uno.
2. El sistema le muestra al usuario una pantalla con los datos del evento del calendario que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda el proceso editado y actualiza el listado de procesos. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.3.3.3. Borrar Evento del Calendario

Diagrama Véase diagrama A.9 en la página 127

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar un evento del calendario existente del sistema.

Precondiciones El administrador terminó el caso de uso “Calendario”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un evento del calendario y elige la opción “Borrar”. Si el usuario no selecciona un evento del calendario, el sistema le avisa que debe seleccionar uno.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra el evento del calendario del sistema y actualiza el listado de evento del calendario.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.4. CRUD Proyectos

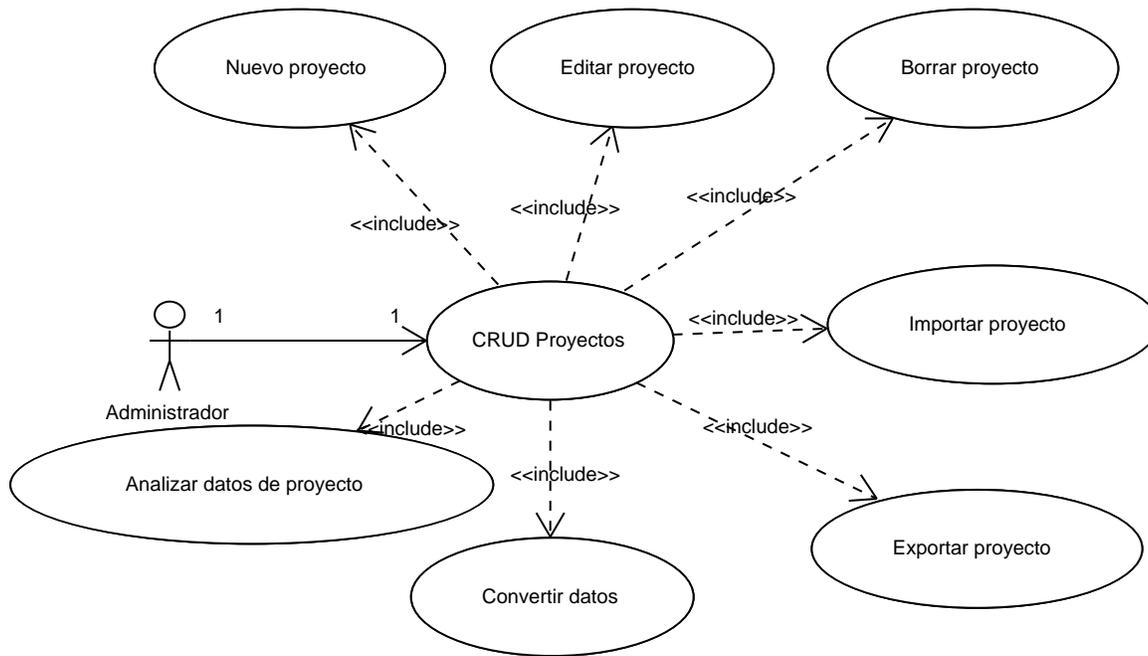


Figura A.10.: Caso de uso de proyectos

Diagrama

Actor primario Usuario, Administrador

Actor secundario Ninguno

Descripción El usuario desea hacer alguna operación con información relacionada con los proyectos

Precondiciones El administrador se encuentra autenticado como tal.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Proyectos”.
2. El sistema le muestra al usuario un menú con los proyectos que tiene registrados el sistema, y un conjunto de acciones que puede realizar con éstos.
3. Si desea agregar un proyecto o importar un proyecto, aquí se puede detener el caso de uso y saltar al caso de uso “agregar proyecto”.
4. El usuario selecciona un proyecto.
5. El sistema le despliega una panel con el detalle del proyecto al usuario.
6. El usuario selecciona una de las opciones para continuar con las acciones.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.4.0.4. Seleccionar proyecto

Diagrama Véase diagrama A.10 en la página anterior

Actor primario Usuario

Actor secundario Ninguno

Descripción El usuario desea seleccionar un proyecto para poder manipular los datos del proyecto con otra funcionalidad del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un proyecto y elige la opción “Hacer activo”. Si el usuario no selecciona un proyecto, el sistema le avisa que debe seleccionar una.
2. El sistema pone como activo el proyecto para esa sesión de usuario y permite que las interacciones con los datos de ese proyecto.

Post-condición El sistema queda con un proyecto seleccionado, al cual se le aplicarán las acciones subsiguientes en el sistema. El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.4.0.5. Añadir proyecto

Diagrama Véase diagrama A.10 en la página anterior

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea agregar un proyecto nuevo al sistema.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Añadir proyecto”.
2. El sistema le muestra al usuario una pantalla con los datos que tiene que llenar.
3. El usuario llena los datos.
4. El sistema valida que los datos estén completos, guarda el nuevo proyecto, actualiza el listado de proyecto y pone como activo el nuevo proyecto. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién creado queda como proyecto activo en el sistema.

A.4.0.6. Editar proyecto

Diagrama Véase diagrama A.10 en la página 130

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea editar un proyecto existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un proyecto y elige la opción “Editar tipo”. Si el usuario no selecciona un proyecto, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla con los datos del proyecto que seleccionó para editar.
3. El usuario modifica los datos.
4. El sistema valida que los datos estén completos y sean correctos, guarda el proyecto editado y actualiza el listado de proyectos. Si no están correctos los datos regresa al usuario al paso 3.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.4.0.7. Borrar proyecto

Diagrama Véase diagrama A.10 en la página 130

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea borrar un proyecto existente del sistema.

Precondiciones El administrador terminó el caso de uso “CRUD Proyecto”.

Flujo de eventos normales y excepcionales

1. El usuario selecciona un proyecto y elige la opción “Borrar proyecto”. Si el usuario no selecciona un proyecto, el sistema le avisa que debe seleccionar una.
2. El sistema le muestra al usuario una pantalla de confirmación.
3. El usuario decide continuar.
4. El sistema borra el proyecto del sistema y actualiza el listado de proyectos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

A.4.0.8. Convertir datos

Diagrama Véase diagrama A.10 en la página 130

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea guardar los datos de un proyecto del sistema al data warehouse.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Convertir datos”.
2. El sistema borra los datos previamente guardados en el datawarehouse del proyecto seleccionado.
3. El sistema guarda los datos del proyecto seleccionado al data warehouse.
4. El sistema muestra una pantalla de confirmación al terminar de convertir los datos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién convertido queda como proyecto activo en el sistema.

A.4.0.9. Exportar datos

Diagrama Véase diagrama A.10 en la página 130

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea exportar los datos de un proyecto del sistema a un archivo.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Exportar”.
2. El sistema exporta los datos del proyecto seleccionado a un archivo XML.
3. El sistema envía al usuario el archivo XML y muestra una ventana para que el usuario elija el lugar donde guardará el archivo en su disco duro.
4. El usuario elige la ruta y el nombre de archivo.
5. Al terminar de exportar, el sistema muestra una pantalla de confirmación de que terminó de exportar.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién exportado queda como proyecto activo en el sistema.

A.4.0.10. Importar datos

Diagrama Véase diagrama A.10 en la página 130

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea importar los datos de un proyecto de un archivo al sistema.

Precondiciones El administrador terminó el caso de uso “CRUD proyectos”.

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Importar”.
2. El sistema le muestra al usuario un formulario en donde puede seleccionar el archivo XML que desea que el sistema importe.
3. El usuario elige la ruta y el nombre del archivo XML que desea importar y oprime continuar.
4. El sistema recibe del usuario el archivo XML y realiza la importación y sincronización de los datos.
5. Al terminar, el sistema muestra una pantalla de confirmación de que terminó de importar los datos.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas. El proyecto recién importado queda como proyecto activo en el sistema.

A.5. Caso de uso Entrada de datos

A.5.1. Llenar registro

Diagrama Ver diagrama A.11 en la página siguiente

Actor primario Usuario

Actor secundario Ninguno

Descripción El usuario puede añadir un nuevo registro de datos a un proyecto del sistema

Precondiciones El usuario se encuentra autenticado como tal.
El usuario tiene activo (seleccionado) un proyecto.

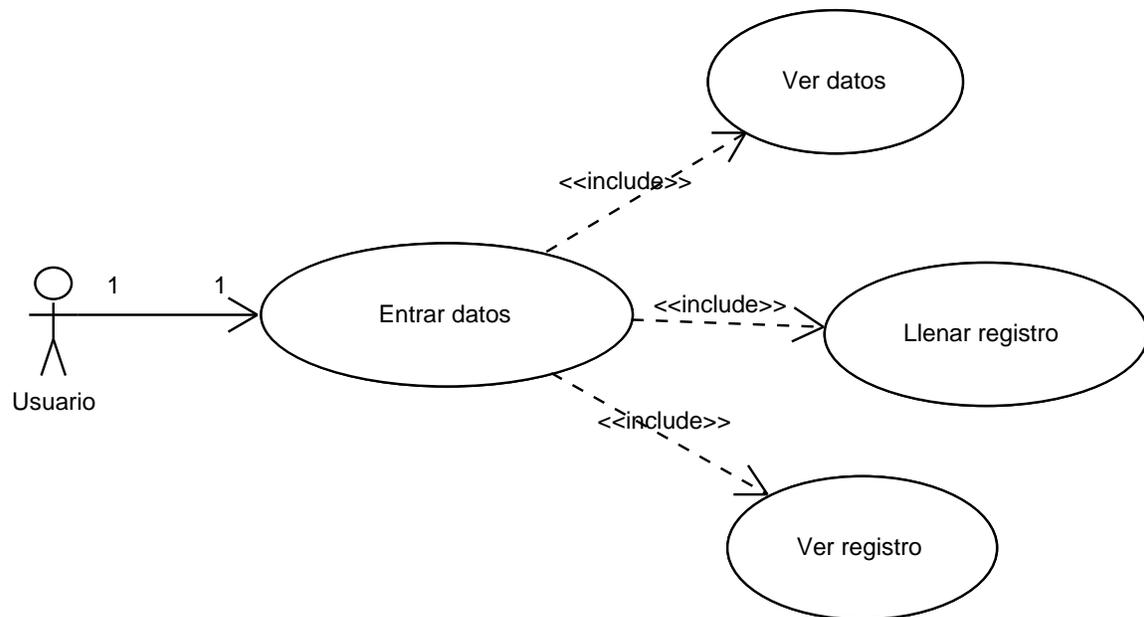


Figura A.11.: Caso de uso entrada de datos

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Respuestas”.
2. El sistema le muestra al usuario un menú con los posibles cuestionarios que puede responder.
3. El usuario selecciona un cuestionario de la lista.
4. El sistema le despliega el cuestionario para que el usuario llene los datos.
5. El usuario llena los datos requeridos por el sistema.
6. El sistema valida y guarda los datos, en caso de haber algún faltante, el sistema regresa al paso 4.

Postcondición El proyecto activo tiene una nueva medición en su base de datos, la cual podrá ser usada en análisis posteriores.

A.5.2. Ver registros

Diagrama Véase diagrama A.11

Actor primario Administrador

Actor secundario Ninguno

Descripción El administrador desea visualizar los datos crudos que residen en el sistema.

Precondiciones El usuario se encuentra autenticado como tal.

A. Casos de Uso

Flujo de eventos normales y excepcionales

1. El usuario elige la opción “Ver datos”.
2. El sistema le muestra al usuario una pantalla con todos los cuestionarios respondidos que tiene almacenado, agrupados por proyecto y fecha.

Post-condición El usuario puede continuar haciendo cualquier acción incluida en este diagrama de uso o sus subdiagramas.

B. La hoja misteriosa

Entre las hojas de reciclaje en mi casa encontré la siguiente que se me hizo muy interesante, seguramente es una fotocopia de algún libro que usó mi hermana en su licenciatura en Psicología. La hoja está demasiado manchada para escanearla (de hecho la escanéé y no se ve bien), así que mejor decidí reproducirla literalmente.

Aunque no hay nombre del libro debido a que estaba en la hoja contraria a la fotocopia sobreviviente, si hay uno del capítulo: “Capítulo 4: Muestreo e investigación de datos impresos”.

...emas. Esto puede o no influenciar el contenido, pero como resultado los datos archivados son una fuente de información de relativo bajo costo. Otra ventaja de los datos archivados es que no hay cambio abierto de los datos, debido a que su productor no está consciente de que está siendo estudiado.

Una desventaja del uso de datos archivados es que el analista puede tener incertidumbre acerca de su significado si solamente existe un subconjunto limitado de los datos originalmente creados. Otra desventaja del uso de datos archivados es que los registros que sobreviven (o que alguien decidió conservar) pueden *no* ser los más importantes o significativos. Una tercera desventaja del uso de datos archivados es que hay un ascendencia integrada, debido a que alguien ha decidido, por la razón que fuere, archivar los datos originales así como conservar algunos datos y no otros. Adicionalmente, es difícil obtener nuevos datos de las mismas muestras o equivalentes. Las ventajas y desventajas del uso de datos archivados se resumen en la figura 4.11 (falta).

Aun con todas las precauciones anteriores, hay algunos lineamientos que pueden hacer valiosa la recopilación de datos archivados. Ellos son:

- Fragmente los datos en subclases y haga revisiones cruzadas para reducir errores.
- Compare reportes sobre el mismo fenómeno por diferentes analistas (o tal vez hasta por alguien de dentro de la organización).
- Dése cuenta de la ascendencia inherente asociada con las decisiones originales para archivar, conservar y/o destruir reportes.
- Use otros métodos, tales como las entrevistas y la observación, para completar su imagen organizacional y proporcionar una revisión cruzada.

La hoja llega hasta aquí. Creo que hay partes que conviene resaltar, entre ellas acerca de que los datos archivados tal vez no sean los más importantes o significativos del lote de datos capturados originalmente, y del hecho que al haber sido archivados selectivamente, necesariamente implica que al que los guardó se le hicieron más importantes, y esa intención sería bueno conocerla para poder hacer un mejor uso de los datos.

C. Compatibilidad con dispositivos móviles

La arquitectura de la aplicación permite visualizarla en dispositivos móviles, ayudando a reducir los requisitos para accederla, permitiendo que miembros de los equipos de proyecto puedan alimentarla usando el acceso a internet de un celular o una PDA.

Una PDA (personal digital assistant) es una computadora del tamaño de la mano. Las PDA más modernas tienen pantallas a color, capacidad de reproducción de audio y tecnología touch screen lo que les permite que puedan ser usadas como teléfonos móviles (smartphone), navegadores de internet o reproductores multimedia portátiles. Muchas PDA pueden acceder a Internet, via redes inalámbricas ya sea Wi-Fi o celular (WWAN).

Un smarphone es un teléfono celular que ofrece características avanzadas, con funcionalidad comparable a una PC.

La mayoría de los dispositivos actualmente considerados como smartphones usan un sistema operativo identificable, normalmente con la capacidad de añadir aplicaciones (para procesamiento de datos, conectividad o entretenimiento), en contraste a los teléfonos normales que sólo soportan aplicaciones en un entorno controlado (sandbox) como juegos Java. Estas aplicaciones para el smartphone pueden ser desarrolladas por el fabricante del dispositivo, por el operador de red o por un tercero dado que el sistema operativo es abierto.

En términos de capacidades, la mayoría de los smartphones combinan el uso de email con un organizador personal. Otras funcionalidades pueden incluir un teclado QWERTY miniatura, touchscreen o teclado numérico, cámara fotográfica, manejo de contactos, acelerómetro, hardware y software de geolocalización integrado, la habilidad para leer documentos de negocios en formatos como PDF o Microsoft Office, software para reproducir música, fotos y video, un navegador de internet o acceso seguro al email de la compañía. Una característica de la mayoría de los smartphones es una lista de direcciones capaz de almacenar tantos contactos como permita la memoria disponible, a diferencia de los teléfonos normales que tienen un número máximo de contactos que se pueden almacenar.

Dado que los dispositivos anteriormente citados difieren en poder de procesamiento, cantidad de memoria, tamaño de pantalla, capacidades de despliegue del navegador de internet, etc. resulta imposible hacer una “versión universal” que pueda visualizarse en cualquier dispositivo portátil o móvil. El reto es crear la aplicación de tal forma que sea relativamente sencillo cambiar la forma de despliegue sin cambiar el código de reglas de negocio.

A continuación se muestran el sistema como se puede ver en un iphone de apple.

Debido a que la interfaz gráfica fue desarrollada para un dispositivo con una pantalla “grande” (al menos 10”) de computadora, el uso del sistema en un sistema con una pantalla de 3” es un poco complicado, requiriendo frecuentes desplazamientos y cambios de tamaño para usar las funcionalidades, sin embargo, la arquitectura subyacente permite, con sólo cambiar el código html desplegado, hacer un nuevo front-end específico, por ejemplo, para un iphone. También en algunos casos las pantallas son demasiado pesadas para un dispositivo de este tipo lo que causa que se tarde demasiado en desplegar, esto se puede corregir separando las pantallas con estos problemas en varias.

El esfuerzo necesario para crear una versión adaptada para un dispositivo móvil específico de la funcionalidad más importante es relativamente bajo. Para ello se tienen que tomar en cuenta las capacidades del dispositivo, cantidad de memoria, tamaño y resolución de pantalla, medios de interacción posibles con el sistema, etc. Es posible que no todas las funcionalidades estén disponibles en cada uno de los dispositivos móviles del mercado debido a los factores antes mencionados. Para realizar la mayoría de las adaptaciones lo más probable es que sólo se necesitaran cambiar los jsp

C. Compatibilidad con dispositivos móviles

que componen el sistema, dejando el resto del código intacto, lo cual supone una cantidad de trabajo relativamente reducida.

El sistema se probó también en otro dispositivo móvil con sistema MS Pocket PC 2005, pero el navegador incluido no soporta la funcionalidad basada en Ajax, aunque las páginas estáticas se despliegan correctamente. Para este caso se necesitaría desarrollar un nuevo front-end que no usara Ajax para que la aplicación se pudiera explotar en estos dispositivos.

C.1. Sistema

El aspecto y la funcionalidad del sistema se puede apreciar en las ilustraciones de la ?? a la C.6 en la página 143.

C.2. Reportes

Los reportes también se pueden visualizar como se apreciar en las figuras C.7 en la página 144, C.8 en la página 144, C.9 en la página 144 y C.10 en la página 144.



Figura C.1.: Página de Bienvenida

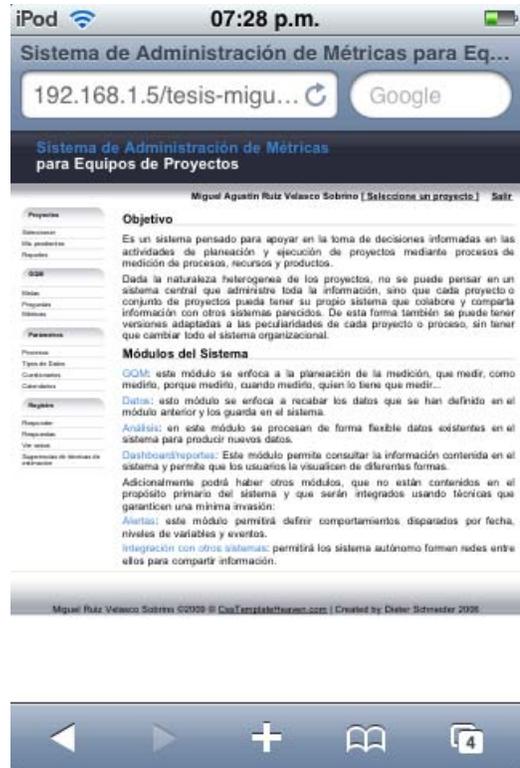


Figura C.2.: Página principal y menú

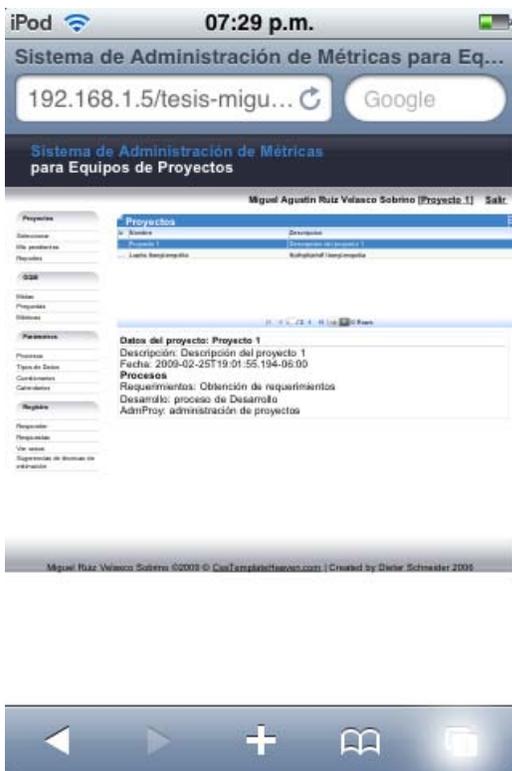


Figura C.3.: Página de Proyectos



Figura C.4.: Página de métricas

C. Compatibilidad con dispositivos móviles

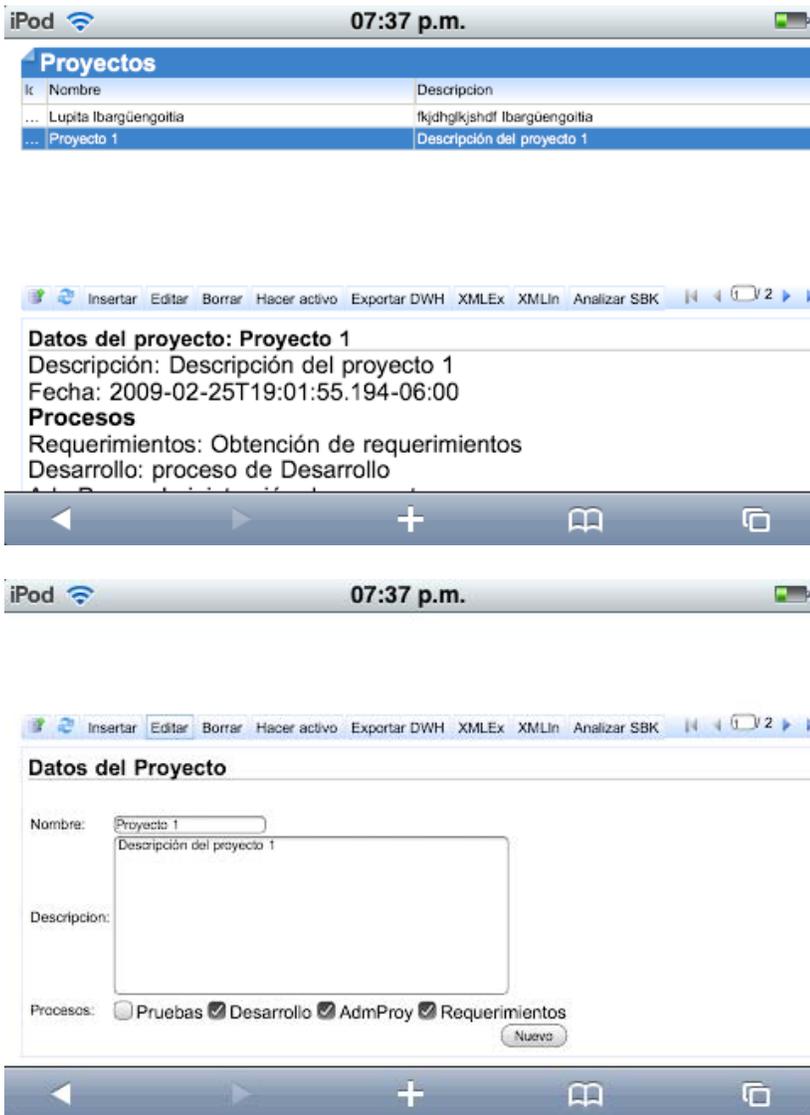


Figura C.5.: Detalle de proyecto

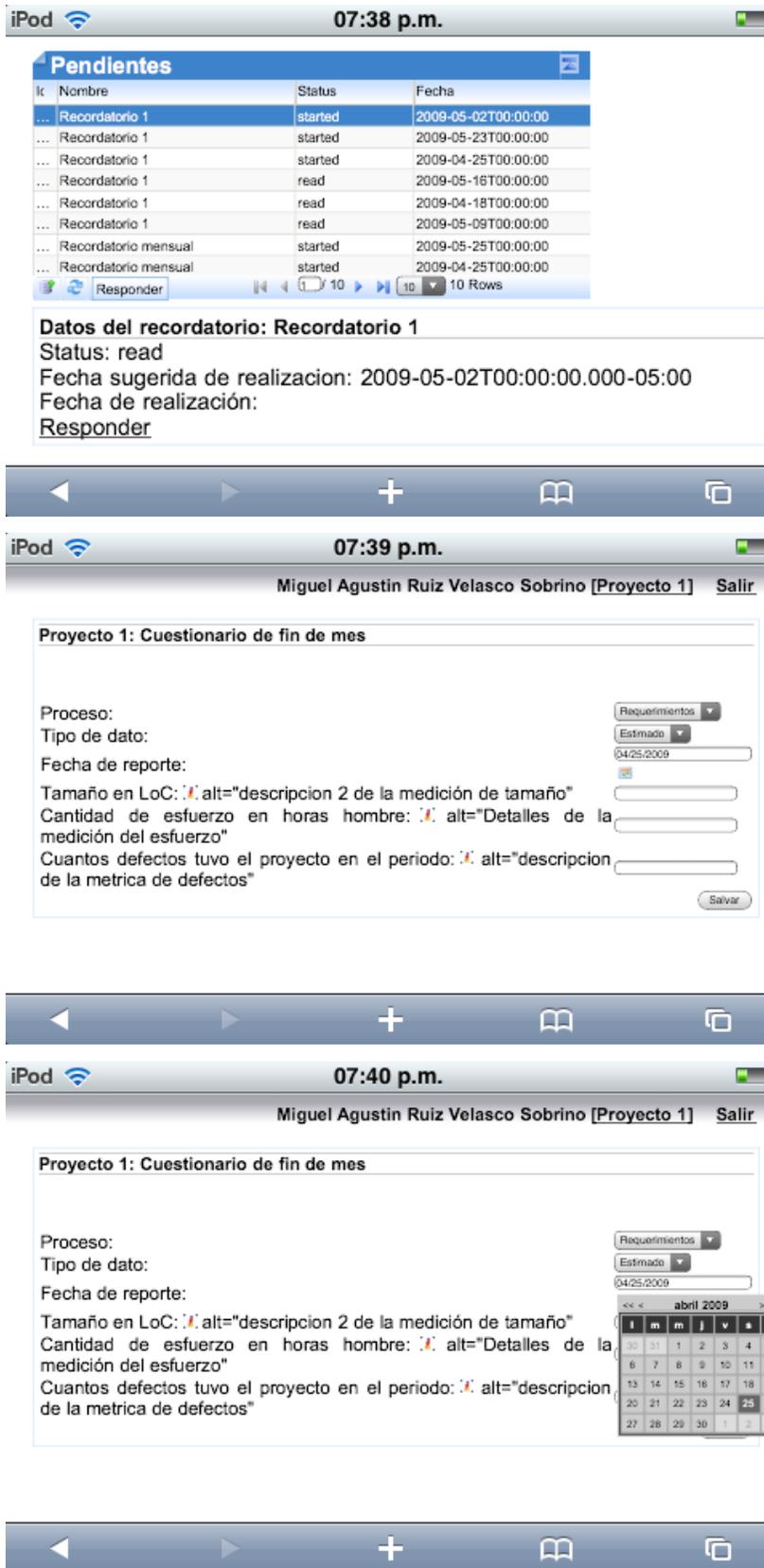


Figura C.6.: Llenado de cuestionario

C. Compatibilidad con dispositivos móviles



Figura C.7.: Entrada a los reportes



Figura C.8.: Página de reportes

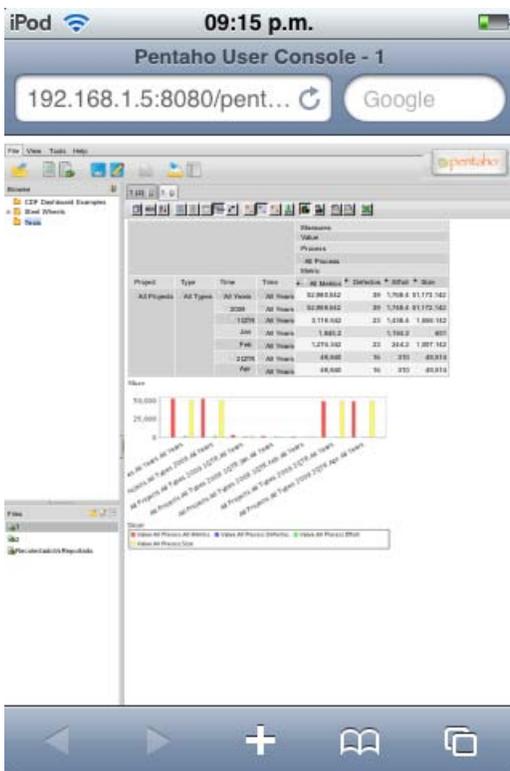


Figura C.9.: Reporte

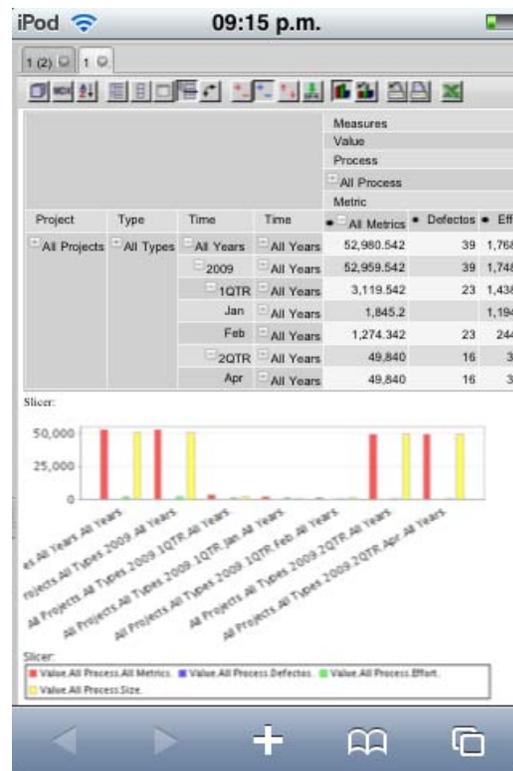


Figura C.10.: Reporte con zoom