



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Métricas para la Interoperabilidad de la Información en el Gobierno Electrónico

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

MAURICIO MONSALVE MORENO

PROFESOR GUÍA:
CLAUDIO GUTIÉRREZ GALLARDO

MIEMBROS DE LA COMISIÓN:
PATRICIO POBLETE OLIVARES
PABLO GONZÁLEZ JURE

SANTIAGO DE CHILE
MARZO 2007

Resumen

La interoperabilidad de la información es un requisito indispensable para un gobierno electrónico funcional, con organismos que trabajan e interactúan a través de medios digitales.

En esta memoria se realizó una revisión de carácter comprensivo de varios temas relacionados al gobierno electrónico. Se da especial énfasis al aspecto de la administración pública: se investigaron tendencias, criterios de evaluación de proyectos y políticas públicas, modelos de análisis de la implementación del gobierno electrónico y características de las instituciones públicas. En particular, se desarrolló un modelo conceptual que indica cuan automatizable es el trabajo en las instituciones públicas.

Se construyó un modelo matemático para modelar y medir aspectos de la interoperabilidad de los servicios web. Partiendo de los fenómenos económicos de una red, se modeló a las redes de servicios web como grafos dirigidos, cuyos nodos son servicios web y sus arcos corresponden a las llamadas entre estos servicios. Y se desarrolló una serie de métricas en base a medidas estadísticas y aspectos de la red modelada, tomando en consideración criterios de gestión de redes informáticas, calidad de servicio y economía de los servicios públicos. Las métricas desarrolladas están dirigidas a orientar las estrategias de mejoramiento de la calidad del servicio, sirviendo además como apoyo para la evaluación del impacto de las mejoras.

Se realizaron tres casos de estudio diferentes. El primer caso de estudio consistió en la realización de una aplicación web que recibe el modelo de una red de servicios web y obtiene ciertas métricas a partir de la información suministrada. Con esta aplicación se realizaron varias pruebas teóricas sobre las métricas. El segundo caso de estudio, de orientación práctica, consistió en el análisis de la situación del Servicio Nacional de Aduanas. El tercer caso de estudio consistió en la comparación del modelo desarrollado con otros trabajos relevantes.

Los modelos desarrollados son especialmente útiles como guías para que los organismos públicos logren y mantengan un buen nivel de servicio a nivel electrónico (servicios web). De esta manera, se espera apoyar el desarrollo de un buen gobierno electrónico. Sin embargo, actualmente existe un impedimento en la aplicación de estos modelos: el escaso desarrollo informático de algunos organismos. Si bien los modelos pueden servir para apoyar organismos inexpertos, la carencia de un área de desarrollo de software es un impedimento para el desarrollo del gobierno electrónico. Al respecto, se realiza una discusión breve acerca de cómo resolver este problema.

Agradecimientos

A mi familia, por su constante apoyo.

A Claudio Gutiérrez, por aguantar a este alumno tan consumidor de tiempo.

A todos aquellos que hicieron posible esta memoria, especialmente a Cristián Vivanco de Aduanas.

Esta memoria fue financiada por el proyecto Fondecyt Procesamiento y Análisis Semántico de Servicios Web, N° 1050642.

Índice general

Agradecimientos.....	iii
Notación.....	xii
1 Introducción.....	1
2 Objetivos de la memoria.....	3
3 Marco teórico.....	5
4 Modelo de análisis del gobierno electrónico.....	38
5 Métricas para las redes de servicios web.....	51
6 Extensiones al modelo.....	86
7 Casos de estudio.....	97
8 Comentarios y conclusiones.....	116
9 Discusión y propuestas.....	117
10 Bibliografía.....	120
Apéndice A: Tecnologías disponibles (énfasis técnico).....	125
Apéndice B: Concentración de la red.....	141
Apéndice C: Estimación del valor del tiempo.....	143
Apéndice D: Códigos de la aplicación.....	144

Índice detallado

Agradecimientos.....	iii
Notación.....	xii
1 Introducción.....	1
1.1 Generalidades.....	1
1.2 Motivación.....	2
2 Objetivos de la memoria.....	3
2.1 Objetivo general.....	3
2.2 Objetivos específicos.....	3
3 Marco teórico.....	5
3.1 Introducción.....	5
3.2 El gobierno electrónico.....	5
3.2.1 La promesa de la tecnocracia.....	5
La Sociedad de la Información.....	6
Economía Digital y Nueva Economía.....	7
La Nueva Gestión Pública (NGP).....	7
La Gobernanza Pública.....	8
3.2.2 Situación internacional.....	8
Canadá.....	9
Corea del Sur.....	9
Estados Unidos.....	9
Estonia.....	9
Filipinas.....	9
La India.....	10
Reino Unido.....	10
Singapur.....	10
3.2.3 Marcos conceptuales.....	11
ITPOSMO.....	11
Modelo de Tecnología habilitante.....	12
Modelo Holístico de Wimmer.....	13
3.3 Las tecnologías disponibles.....	13
3.3.1 El formato documental XML.....	13
Esquema de un documento XML.....	15
Manipulación de documentos XML.....	16
3.3.2 Los servicios web.....	17
Arquitectura orientada al servicio: SOA.....	17
Los estándares de la W3C para los servicios web.....	18
3.4 La evaluación socio-económica.....	18
3.4.1 Criterios de aprobación de proyectos y políticas.....	19
Criterio de Pareto.....	19
Criterio de Kaldor-Hicks.....	20

Criterio de Bergson-Samuelson.....	20
Otros criterios.....	21
3.4.2 La teoría de la evaluación social de proyectos.....	21
Enfoque de eficiencia.....	22
Enfoque distributivo.....	22
3.4.3 La práctica de la evaluación social.....	23
El valor presente neto.....	23
Las distorsiones del mercado.....	25
3.4.4 Indicadores de evaluación.....	25
IVAN (criterio).....	25
Indicadores Costo-Beneficio.....	26
Indicadores Costo-Impacto.....	27
3.5 Las métricas en la ingeniería de software.....	27
3.5.1 Problemas de las métricas.....	27
3.5.2 Métricas en casos específicos.....	29
Métricas de módulos prefabricados (COTS).....	30
Modelo COCOMO.....	31
Cuadro de Mando Integral.....	31
Metodología Goal-Question-Metric (GQM).....	32
Métricas bayesianas.....	33
3.5.3 Definiciones formales.....	34
3.6 Resumen.....	36
4 Modelo de análisis del gobierno electrónico.....	38
4.1 Introducción.....	38
4.2 La administración pública actual.....	38
4.3 Tendencias en la modernización del estado.....	39
4.4 Metodología general de implementación.....	41
4.5 Una clasificación para los flujos de información.....	42
4.5.1 Clasificación del tipo de documentos según la naturaleza del trabajo.....	42
4.5.2 Clasificación según la naturaleza programable del trabajo.....	44
4.5.3 Implementación del gobierno electrónico según la naturaleza del trabajo.....	45
4.6 La situación real y la situación ideal.....	45
4.7 Un esquema general.....	48
4.8 Resumen.....	49
5 Métricas para las redes de servicios web.....	51
5.1 Introducción.....	51
5.2 Consideraciones preliminares.....	51
5.2.1 Bienestar social y calidad del servicio.....	51
5.2.2 Función de provisión del servicio electrónico.....	52
5.2.3 Efectos económicos de red.....	53
5.3 Modelo de la red de servicios web.....	55
5.3.1 La red de dependencias.....	55
5.3.2 Requerimientos de calidad de la red.....	59
5.4 Métricas y calidad de servicio.....	62

5.4.1 Controlando el nivel de servicio.....	62
5.4.2 Midiendo el nivel de servicio.....	63
5.4.3 Métricas estructurales.....	66
5.4.4 Métricas de funcionamiento.....	70
5.5 Modelamiento de una red.....	76
5.5.1 Modelamiento desde una situación real.....	76
Construcción de la estructura de la red.....	76
Alimentación de las métricas.....	77
El caso de una red limitada.....	77
5.5.2 Modelamiento desde una situación hipotética.....	78
Diseño de una red.....	78
Métricas a usar.....	78
Obtención de la demanda interna o dependiente.....	79
5.6 Utilización de las métricas.....	80
5.6.1 Estrategias para mejorar el nivel de servicio.....	80
5.6.2 Evaluación del impacto de una mejora.....	81
Beneficio social debido a una reducción del tiempo de servicio.....	81
Impacto debido al menor uso de la red.....	84
5.7 Resumen.....	84
6 Extensiones al modelo.....	86
6.1 Introducción.....	86
6.2 Extensión: Tolerancia a ciclos.....	86
6.2.1 El problema.....	86
6.2.2 Cambios en las métricas.....	87
6.3 Extensión: red de grupos.....	90
6.3.1 Definiciones.....	90
6.3.2 Conversión de una red de servicios web a una red de grupos.....	91
6.3.3 Construcción de una red de grupos como un diseño.....	93
6.3.4 Métricas aplicables a la red de grupos.....	94
6.4 Extensión: Nodos de distinta naturaleza.....	95
6.5 Comentarios.....	96
7 Casos de estudio.....	97
7.1 Pruebas teóricas y desarrollo.....	97
7.1.1 Aplicación desarrollada.....	97
7.1.2 Utilización de la aplicación.....	97
7.1.3 Pruebas teóricas.....	100
Concentración y desconcentración de la red.....	100
Simulación de un análisis.....	101
7.2 Caso práctico: Aduanas.....	104
7.2.1 La modernización de Aduanas.....	104
7.2.2 Redes de servicios web en Aduanas.....	104
7.2.3 Otras experiencias.....	108
7.3 Comparación con otros desarrollos teóricos.....	110
7.3.1 “Modeling Quality of Service for Workflows and Web Service Processes”.....	110

7.3.2 “An Architecture for QoS Enabled Dynamic Web Service Deployment”.....	111
7.3.3 “An Automata based approach to e-Government cooperation”.....	112
7.3.4 “Research report on interoperability and co-ordination of Web Services”	113
7.3.5 Comentarios.....	114
8 Comentarios y conclusiones.....	116
9 Discusión y propuestas.....	117
10 Bibliografía.....	120
10.1 Libros, artículos e informes.....	120
10.2 Sitios web.....	123
Apéndice A: Tecnologías disponibles (énfasis técnico).....	125
El formato documental XML.....	125
Sintaxis de un documento XML.....	126
Esquema de un documento XML.....	128
DTD: Document Type Definition.....	129
RELAX NG: RELAX y TREX.....	130
XSD: XML Schema.....	131
Namespaces.....	132
Manipulación de documentos XML.....	133
XPath.....	133
XQuery.....	134
XSLT.....	136
Los servicios web.....	138
Arquitectura orientada al servicio: SOA.....	138
Los estándares de la W3C para los servicios web.....	139
XML.....	139
SOAP.....	139
WSDL.....	139
WS-CDL.....	140
Apéndice B: Concentración de la red.....	141
Concentración y desconcentración de la red.....	141
Obtención de las fórmulas.....	141
Apéndice C: Estimación del valor del tiempo.....	143
Apéndice D: Códigos de la aplicación.....	144

Índice de figuras

Figura 1: Modelo de la brecha en ITPOSMO.....	12
Figura 2: Modelo de tecnología habilitante de Fountain.....	12
Figura 3: Modelo holístico de Wimmer.....	13
Figura 4: Criterio de Pareto.....	19
Figura 5: Atributos y métricas [Fenton 2000].....	28
Figura 6: Métricas de nivel de sistema para COTS [Sahra 2001].....	30
Figura 7: Modelo de red bayesiana para estimar atributos de un proyecto [Fenton 2000].....	33
Figura 8: Documentación según estructura y decisiones.....	43
Figura 9: Naturaleza programable del trabajo.....	44
Figura 10: Apoyo tecnológico según la naturaleza del trabajo.....	45
Figura 11: Brecha entre diseño y realidad (Heeks).....	46
Figura 12: Estado actual y estado ideal.....	47
Figura 13: Mapa conceptual simplificado.....	48
Figura 14: Bienestar social y calidad del servicio.....	52
Figura 15: Crecimiento, complejidad y reducción de costos.....	54
Figura 16: Dependencias directas e indirectas.....	56
Figura 17: Dependencia mixta.....	59
Figura 18: Dependencia indirecta y eficiencia.....	59
Figura 19: Entrantes y salientes.....	67
Figura 20: Concentración y desconcentración.....	68
Figura 21: Potenciales llamadas redundantes.....	70
Figura 22: Nodos y arcos: servicios web y llamadas.....	76
Figura 23: Estrategias genéricas para el nivel de servicio.....	80
Figura 24: Evaluación socio-económica (ejemplo).....	84
Figura 25: Un grafo con un ciclo dirigido.....	87
Figura 26: Concepto de red de grupos.....	91
Figura 27: Una red agrupada sobre una red desagrupada.....	92
Figura 28: Red con servicios web y base de datos.....	96
Figura 29: Ingreso de la red de dependencias.....	98
Figura 30: Menú de trabajo.....	98
Figura 31: Visualización de la red (forma centrada).....	99
Figura 32: Cálculo de CargaSGen.....	99
Figura 33: Visualización detallada.....	100
Figura 34: GrConcen y GrDesconcen de una lista.....	101
Figura 35: Visualización y métricas globales.....	102
Figura 36: Cálculo de CargaSGen.....	102
Figura 37: Cálculo de CargaRGen.....	103

Figura 38: Cálculo de la Demanda Interna.....	103
Figura 39: Ventanilla Única de Comercio Exterior.....	105
Figura 40: Conexión con Sernapesca.....	106
Figura 41: Propuesta de conexión.....	107
Figura 42: Interoperación entre ISP, Aduana, Tesorería y la banca.....	108
Figura 43: Un paso en la normalización del grafo.....	111
Figura 44: Composición de servicios web como autómatas.....	113
Figura 45: Modelo de flujo de trabajo, QUALEG.....	114
Figura 46: Árbol de un documento XML.....	128

Índice de códigos

Código 1: Ejemplo XML sencillo.....	14
Código 2: Dos entidades <texto> dentro de <parrafo>.....	15
Código 3: Cálculo de la demanda de los grupos.....	92
Código 4: Cálculo de Pr y Nl en la red de grupos.....	93
Código 5: Formato de ingreso de una red.....	101
Código 6: Ejemplo XML sencillo.....	126
Código 7: Anidamiento en XML.....	126
Código 8: Dos entidades <texto> dentro de <parrafo>.....	126
Código 9: Formas equivalentes.....	127
Código 10: Inclusión de un atributo.....	127
Código 11: Documento XML con aridad.....	127
Código 12: Un DTD sencillo.....	130
Código 13: Ejemplo sencillo de RELAX NG.....	131
Código 14: Ejemplo de XML Schema.....	132
Código 15: Ejemplo de namespaces.....	133
Código 16: Árboles y frutos.....	134
Código 17: FLOWR (XQuery).....	135
Código 18: Consulta FLOWR.....	135
Código 19: Consulta FLOWR con formato HTML.....	135
Código 20: Árboles y frutos (repetido).....	136
Código 21: Transformación XSLT.....	137

Notación

\wp : Probabilidad. Ej. $\wp(x=0)$ es la probabilidad que $x=0$.

σ : Desviación estándar.

\mathbb{R} : Números reales.

\mathbb{N} : Números naturales (sin cero).

\mathbb{N}_0 : Números cardinales o naturales con cero.

VPN : Valor presente neto.

$G(W, L)$: Grafo cuyos nodos están en W y sus arcos en L .

$\frac{\partial f}{\partial x}$: Derivada parcial. En este caso, la derivada parcial de f con respecto a x .

$\partial_x f$: Notación simplificada de $\frac{\partial f}{\partial x}$.

$\frac{df}{dx}$: Derivada directa. En este caso, la derivada directa de f con respecto a x .

1 Introducción

1.1 Generalidades

El término Gobierno Electrónico (e-government) fue usado por primera vez en diciembre de 1999 por Al Gore, el entonces vicepresidente de Estados Unidos, a través de un memorando del presidente Bill Clinton, titulado “E-Government Directive” [PRYME 2006]. Este memorando instruye a las instituciones a usar las tecnologías de la información para facilitar el acceso a los servicios y la información del Gobierno, de forma que estén al alcance del público. Un segundo memorando, titulado “E-Society Directive”, insta a las instituciones públicas a usar estas mismas tecnologías para mejorar la calidad de vida de los ciudadanos.

La estrategia para el Gobierno Electrónico en Chile ya tiene sus años de historia [DCC-1 2005] [Bravo 2005] [PRYME 2006]. En el año 1999 ocurren los hitos de la creación de la Intranet del Estado y el Portal del Gobierno de Chile. En el año 2000 se crea el Proyecto de Reforma y Modernización del Estado, el PRYME. En el año 2001, aparecen las directrices para la estrategia de Gobierno Electrónico en el Instructivo Presidencial. En el año 2002, la ley de documento y firma electrónica es aprobada. Y en el año 2004 aparece el Decreto Supremo N° 81, el decreto que fija el formato de la documentación electrónica en XML.

¿Por qué fijar un formato para la documentación electrónica? Para la interoperabilidad de los sistemas a través de este formato [Klischewski 2004]. Se entiende por interoperabilidad como “capacidad, conocimiento y acuerdo de dos o más partes de un todo para interoperar” [Bravo 2005]. Por ende, la interoperabilidad es necesaria para que el Gobierno Electrónico actúe como tal, funcionando como una unidad salida de la suma de las partes, respetando el funcionar de las instituciones.

Es muy importante medir el progreso y la calidad de la interoperabilidad de la información en el Gobierno Electrónico. Respecto a esto, se ha manifestado un quiebre que dificulta el progreso. Citando literalmente el informe de avance del Gobierno Electrónico en Chile [PRYME 2006], página 124, sección 5.6: “Primero, podemos señalar que no existe hasta la fecha un modelo integrador que dé cuenta del marco teórico conceptual de cómo evaluar el avance en Gobierno Electrónico. A modo de comentario, este modelo tampoco ha sido desarrollado a nivel internacional, pero sí existen una serie de aportes en la medición de aspectos aislados, que han permitido avanzar en esta línea.” Es por eso que ha sido incluido en uno de los desafíos (Desafío 2, p. 133 [PRYME 2006]): “desarrollar y utilizar un enfoque integrador de evaluación de las iniciativas de Gobierno Electrónico.” Por esta razón, el tema de esta memoria es el desarrollo de un modelo de evaluación del Gobierno Electrónico, centrado en el aspecto de la interoperabilidad

de la información.

1.2 Motivación

La principal motivación de este tema de memoria es su relevancia para el desarrollo del país. Esto es, realizar un aporte a lo que es la comunicación con la ciudadanía, mayor democracia, transparencia institucional y gubernamental, trámites más eficientes, mejores servicios, mejores decisiones a nivel de gobierno y mayor efectividad en el desarrollo de las iniciativas de éste.

Desde el punto de vista académico, el tema de las métricas de la interoperabilidad en el Gobierno Electrónico es muy interesante. Primero, recae un interés sobre las métricas por su importancia para la Ingeniería de Software. Segundo, la interoperabilidad en el Gobierno Electrónico es un tema que está abordando el Departamento de Ciencias de la Computación. Tercero, existen pocos trabajos sobre métricas en el Gobierno Electrónico. Esto es mucho menor si el área de observación es Latinoamérica. Cuarto, no existe a nivel internacional un modelo integrador para medir y evaluar el progreso en el Gobierno Electrónico [PRYME 2006]. Debido a la necesidad de implementar uno localmente, es posible usar esta memoria para satisfacer esta necesidad.

El desafío, desde el punto de vista técnico, surge de realizar estudios y prácticas en los temas de interoperabilidad de sistemas, XML, Web Semántica y el desarrollo de las métricas. En particular, se vislumbran muchos desafíos en cuanto al XML y la Web Semántica en el Gobierno Electrónico [DCC-1 2005] [DCC-2 2005]. En cuanto a las métricas, ha existido gran dificultad en el establecimiento de objetividad y precisión [Kaner 2004]. Además, es interesante el hecho que en el Gobierno Electrónico confluyen temas como computación, economía, política, sociología, etc. Entonces, además del desafío técnico, existe un desafío de ámbito. La digitalización de las interacciones sociales está generando nuevos fenómenos como la Democracia Electrónica y la Ciudadanía Electrónica [Stamoulis 2001]. Estos temas están cubiertos en la Gobernabilidad Electrónica, tema en el cual “no hay claridad de cómo avanzar” [PRYME 2006].

Finalmente, se puede estimar que un desarrollo en esta área provocará gran impacto. Primero, el impacto se verá en el Gobierno de Chile. El Proyecto de Reforma y Modernización del Estado será el cliente directo. Transitivamente, el resto del Gobierno, las instituciones y finalmente la ciudadanía serán partícipes del impacto de este desarrollo [PRYME 2006]. Segundo, esta memoria será pionera en el tema de métricas en la interoperabilidad de la información en el Gobierno Electrónico en Chile [PRYME 2006]. Tercero, dependiendo del éxito del trabajo desarrollado, otros países en vías de desarrollo serán eventuales clientes de los resultados de esta memoria.

2 Objetivos de la memoria

2.1 Objetivo general

La construcción de una serie de métricas y criterios de decisión que entreguen información útil para la gestión y evaluación de proyectos de interoperabilidad de Gobierno Electrónico. Respecto a las métricas, éstas representarán cuantitativamente factores elementales de la interoperabilidad (i.e. factores de calidad), características de los proyectos informáticos (i.e. métricas de los procesos) o serán alguna derivación de las anteriores. Los criterios de decisión tomarán base en las métricas para responder preguntas de gestión de proyectos informáticos de interoperabilidad, tales como la elección de realizar o no un proyecto, si la calidad alcanzada es suficiente, si están ocurriendo problemas, etc.

Estas herramientas deben funcionar de forma práctica y efectiva. Las mediciones deberán ser fáciles de realizar y deberán asegurar algún grado de exactitud. Naturalmente, estas herramientas deben estar sustentadas por una teoría que les dé fundamentos sólidos.

2.2 Objetivos específicos

MARCO CONCEPTUAL DE ANÁLISIS DE LA INTEROPERABILIDAD: Construir un marco conceptual para el análisis del Gobierno Electrónico en su aspecto de la interoperabilidad.

DISTINGUIR ATRIBUTOS MEDIBLES: Identificar aquellos aspectos del Gobierno Electrónico que son medibles de forma sencilla, barata y efectiva, casi de manera elemental¹, sin que la medición se estropee por temas subjetivos.

MODELO FORMAL PARA LAS MÉTRICAS: Construir un modelo matemático (al menos) que interrelacione las métricas y les dé sustentos sólidos². Así se pretende un enfoque de medición objetivo y cuantitativo, tratando de evitar la intervención subjetiva en éstas.

APOYAR LA INGENIERÍA DEL SOFTWARE: Se pretende apoyar la labor de la gestión de proyectos informáticos de interoperabilidad de Gobierno Electrónico a través de una serie de métricas que entreguen información sobre riesgos y estado de avance de éstos.

1 Una medición elemental se llama métrica directa, y se asume por válida [Kaner 2004].

2 Por definición, una métrica es un mapeo que convierte atributos de objetos o eventos en números o símbolos. Para evitar ambigüedad en el mapeo, se hace necesario basarse en un modelo o teoría [Kaner 2004].

APOYAR LA LABOR DE GESTIÓN: Se planea facilitar la labor de la gestión dentro del proyecto de Gobierno Electrónico, tanto en temas de costos y beneficios como en criterios de decisión para proyectos de complejidad tecnológica.

GENERAR MÉTRICAS LISTAS PARA USAR: Mediante una serie de estudios se pretende lograr que muchas de las métricas se puedan usar inmediatamente, sin que quien las use tenga que ajustar muchos parámetros. Idealmente, se pueden incluir estimaciones de los parámetros de los modelos.

3 Marco teórico

3.1 Introducción

En el gobierno electrónico convergen temas de áreas muy distintas, tales como política, sociología, gestión, computación, economía, antropología, etc. Estas áreas son bastante distintas y cada una realiza un aporte importante al desarrollo de las teorías de gobierno electrónico. Sin embargo, en esta memoria se usarán sólo unas pocas disciplinas.

Este capítulo está dividido en cuatro partes:

1. EL GOBIERNO ELECTRÓNICO: introduce a las tendencias y el estado del arte del gobierno electrónico a nivel internacional.
2. LAS TECNOLOGÍAS DISPONIBLES: introduce a las tecnologías web. Estas tecnologías han catalizado el desarrollo del gobierno electrónico.
3. LA EVALUACIÓN SOCIO-ECONÓMICA: introduce a la evaluación social de proyectos, la metodología por la cual se mide el impacto y el costo de los proyectos sociales.
4. LAS MÉTRICAS EN LA INGENIERÍA DE SOFTWARE: introduce al estado actual de las métricas en la ingeniería de software y al formalismo de las métricas.

3.2 El gobierno electrónico

La revolución del gobierno electrónico se está viviendo en todo el mundo. Su explosión se ha hecho notar desde inicios del 2000. Sin embargo, la tendencia del gobierno electrónico es parte de una tendencia más grande, la modernización de la administración pública, como reacción al “anticuado” sistema burocrático.

3.2.1 La promesa de la tecnocracia

Las tecnologías de la información y las comunicaciones (TIC) llevan bastante tiempo siendo implementadas en las organizaciones privadas (en la industria), con resultados generalmente exitosos. Estas tecnologías han posibilitado la agilización de los procesos de negocios, han aumentado los niveles de comunicación dentro de las empresas, han permitido acercar a los clientes, han permitido mejorar la logística debido al soporte tecnológico de las decisiones, han

simplificado las operaciones contables y financieras, han simplificado la gestión documental, etc. Actualmente, es muy difícil pensar que una empresa tenga buen desempeño si no tiene soporte informático. El uso de las TIC se ha hecho un estándar industrial.

Comparativamente, la administración pública se estaba quedando atrás. La eficiente burocracia se volvió ineficiente. Este y otros problemas han influenciado el ingreso de las TIC como norma en la administración pública (si bien antes habían bastantes iniciativas de uso de las TIC en el gobierno, ahora el uso se ha vuelto obligatorio).

La administración pública ha sido usuaria de las tecnologías de la información hace bastante tiempo. Su uso se remonta hacia 1960 (en Chile hacia los 1980). Sin embargo, la utilización de los computadores era similar al uso de máquinas de escribir [Criado 2002]. La nueva “revolución” se basa en un gobierno que funciona gracias a las TIC. Esa es la promesa de la *interoperabilidad*.

LA SOCIEDAD DE LA INFORMACIÓN

La sociedad tecnocrática -basada en la tecnología- se ha vuelto una realidad. Los avances tecnológicos se masifican rápidamente y se vuelven parte de la cultura de las personas.

La sociedad de la información es un término acuñado en Europa. Este concepto sintetiza la necesidad de plantear políticas de desarrollo, promoción y universalización de las TIC. La Comisión Europea la explica como la “liberalización del sector de las telecomunicaciones, flexibilización de su marco regulador, promoción de las redes sociales y potenciación de los aspectos sociales y culturales” [Criado 2002].

Otro punto de vista, más sociológico, se encuentra en [Rivera 2003]. En éste, se caracteriza a la sociedad de la información como un fenómeno cultural que rompe los paradigmas clásicos, como la diferencia entre animal y máquina, y las diferencias espaciales y temporales (la información atraviesa el globo casi instantáneamente). En la sociedad de la información, la mente humana es una máquina productiva directa. Ello se ve reflejado en la oferta de bienes y servicios, tanto en los ámbitos material e intelectual. Por otra parte, la sociedad de la información se basa en una economía en donde la red es lo importante, más que los individuos en sí. Y estas redes son estructuras abiertas, en las que cualquiera es bienvenido. En este sentido, la nueva cultura no es una cultura en su sentido tradicional. Se podría decir que ésta es una cultura paradójica³, porque

3 La paradoja viene de la concepción del término “cultura”. La cultura es un conjunto de ritos, conocimientos, etc. que caracterizan a una sociedad y que no se transmiten por herencia biológica (en contraste, su herencia se llama *herencia cultural*). Sin embargo, una cultura diversa pasa a ser una cultura de la no-cultura, pues la caracterización del grupo desaparece. Esa es la paradoja.

tiene una cohesión mínima.

ECONOMÍA DIGITAL Y NUEVA ECONOMÍA

El término *economía digital* es el símil norteamericano al término *sociedad de la información* acuñado en Europa [Criado 2002]. La economía digital es un modelo conceptual que privilegia, sobre todo, la mejora del funcionamiento del mercado, que es el que ha de permitir la emergencia de todos los nuevos servicios y actividades que las TIC pueden generar.

La *nueva economía* se basa en ser una economía interconectada [Rivera 2003], que permite una explosión en la productividad. Oferta y demanda se acercan, y la información es más disponible (acercándose a los modelos económicos de competencia perfecta). Sin embargo, esta economía parece ser una cultura del *derroche*: vale más descubrir o crear oportunidades que solucionar problemas [Rivera 2003].

La nueva economía aparece en la explosión de la Internet (el *boom* de las .com). Las expectativas desaparecieron con la crisis económica, originando una gran caída. Esto provocó una fuerte decepción. Muchos consideraron esta revolución económica como un espejismo [Rivera 2003].

Sin embargo, el desarrollo de la Internet sigue, siempre destacando por sus importantes innovaciones.

LA NUEVA GESTIÓN PÚBLICA (NGP)

La Nueva Gestión Pública (NGP) o *New Public Management* (NPM) es una tendencia de administración pública que busca hacerla similar a la administración privada⁴. La NGP busca lograr la eficiencia institucional a través de esto.

La NGP comprende, más o menos, las siguientes dos escuelas de gerencialismo⁵:

- Nueva Economía Institucional⁶: Se basa en la teoría de la elección pública, la teoría de

4 No todas las tendencias de la NGP son así. De todas maneras, el concepto tras la NGP parece ser difícil de sintetizar. De hecho, no hay acuerdo [Criado 2002].

5 Gerencialismo o managerialismo.

6 Son bases teóricas tomadas de la economía. La teoría de los costes es ampliamente conocida -es algo intuitiva- mientras la teoría de la elección pública se basa en el objetivo económico de la administración pública (ver más adelante la sección de evaluación socio-económica) y la teoría de la agencia es una aplicación de la teoría de juegos al diseño de contratos, según la observabilidad, el rendimiento probabilista del *ejecutor* (el agente) y la disposición a pagar. La teoría de la agencia también es conocida

los costes de transacción y la teoría de la agencia.

- Gestión industrial⁷: Las tendencias de gestión privadas, basadas en el control de gestión y la gestión de organizaciones.

El gobierno electrónico, entonces, pasa a ser una implementación tal como en el sector privado. Y sus objetivos principales son la eficiencia y la flexibilidad institucional.

LA GOBERNANZA PÚBLICA

La Gobernanza Pública busca rescatar o preservar los valores democráticos en la gestión pública. Lidia con el problema de que la competencia no siempre permite soluciones adecuadas. La sanción no siempre es la vía. Muchas veces, la cooperación y la coordinación son necesarias, entre el gobierno y una gran cantidad de actores.

La gobernanza comprende dos partes: la gobernanza como tal (*governance*) y la acción de gobierno (*governing*) [Criado 2002]. De acuerdo a esto, los procesos de gobernanza se dividen en dos:

- Una parte *estructural* -el neoinstitucionalismo- que hace referencia en los **arreglos institucionales existentes** en una sociedad dada.
- Una parte *dinámica -policy networks-* que hace referencia a las **acciones de las redes de actores** que pueden afectar la parte estructural.

Los valores democráticos se refieren a la inclusión de todos los actores en los procesos de gobernanza y a la posibilidad de modificar las estructuras existentes.

3.2.2 Situación internacional

A nivel internacional, el gobierno electrónico tiene un desarrollo bastante observado. Esto ha provocado una especie de *carrera tecnológica* hacia la *tecnocracia*.

A continuación se presenta brevemente el desarrollo del gobierno electrónico en algunos países⁸:

como “el agente y el principal”.

⁷ Aquella que aparece en cualquier revista de *management* y de negocios. Es una disciplina muy popular.

⁸ Esta subsección está basada en [PRYME 2006], [infoDev 2002] y [Pascual 2003].

CANADÁ

Canadá ha dispuesto un gran número de servicios en línea, de tipo informativo y de tipo operativo. Este país se destaca especialmente por la participación de los ciudadanos en el proceso de modernización: se han brindado mecanismos de participación para los usuarios opinen y respondan encuestas. Así, los portales se han diseñado tomando en consideración las necesidades de los ciudadanos.

COREA DEL SUR

Corea del Sur ha implementado un sistema unificado de información del gobierno: New Korea Net-Government (NKN-G). Interconecta a través de fibra óptica gobierno central, gobiernos locales, instituciones públicas, universidades y centros de investigación. La provisión de servicios se agilizó y se hizo disponible las 24 horas del día. En cuanto a transparencia, el proyecto OPEN ha permitido a los ciudadanos obtener cualquier tipo de información del gobierno, así como los seguimientos de sus trámites, tanto a nivel local como central.

ESTADOS UNIDOS

En Estados Unidos se ha diseñado una nueva visión para el gobierno electrónico, llamada *Expanded Electronic Government*, que promete ser centrada en el ciudadano. Específicamente, estos desarrollos han tenido los siguientes objetivos: prestación de servicios de calidad, reducción de costos, mayor transparencia y acceso más fácil a los servicios. Se ha dado prioridad a las iniciativas que tienen más impacto y alcance, como los que afectan transversalmente a los servicios públicos (e-adquisiciones, firma digital, etc.), para facilitar la gestión de la cadena de valor en el largo plazo.

Este país tiene varios desafíos importantes en el área de la interoperabilidad. Nótese que Estados Unidos tiene una estructura federal. Cada estado tiene sus leyes. Y cada municipio tiene a su cargo un área espacial importante de un estado.

ESTONIA

En Estonia, el acceso a Internet fue declarado un *derecho humano*. Los planes han sido asegurar el acceso a Internet y otorgar correo electrónico a toda la población. En la misma línea, se realizaron asociaciones entre la industria y las escuelas para proveer acceso a Internet a todas ellas y para producir una generación completa de *letrados digitales*.

FILIPINAS

Filipinas ofrece unos casos interesantes. La recolección de impuestos, realizada por la BIR

(*Boureau of Internal Revenue* u Oficina de Ingresos Internos), se realiza en línea y hace una confirmación por SMS⁹ para prevenir fraudes. En otra iniciativa, la NBI (*National Boureau of Information* u Oficina Nacional de Información) recibía reclamos porque se demoraba tres días en entregar la *NBI Clearance* (algo así como una ficha de antecedentes), que es requerida para postular a trabajos, obtener visas y otras cosas. Actualmente, la operación se realiza en cinco minutos, y se reportaron bajas en la corrupción (no es posible sobornar a una máquina ni mostrarle documentos falsificados), bajas en los costos (tanto del NBI como de los ciudadanos), descongestión de las instalaciones, etc.

LA INDIA

La India ha desarrollado bastantes sistemas de información gubernamentales. Y tiene grandes desafíos, como Estados Unidos, por ser federal. Por ejemplo, se ha dado la posibilidad de pagar los peajes¹⁰ con tarjetas de crédito (un ejemplo de interoperabilidad entre estados). Una iniciativa interesante ha sido la creación de “kioskos electrónicos” (como *infocentros*, pero donde se cobra una tarifa pequeña para el acceso) para las áreas rurales, que otorgan desde servicios de pago de impuestos hasta teleconferencia¹¹, usando pequeños emprendedores.

REINO UNIDO

El Reino Unido ha tenido un compromiso constante con el desarrollo del gobierno electrónico. El desarrollo del gobierno electrónico está a cargo del e-Envoy, el cual tiene dos equipos: un equipo de políticas (Policy Team) y un equipo de entregables¹² (Delivery Team). Destacables han sido los lineamientos técnicos, como el eGIF [eGIF 2005], que han sido íconos en el desarrollo del gobierno electrónico a nivel mundial. También se ha permitido la participación ciudadana para capturar sus necesidades y opiniones.

SINGAPUR

El desarrollo del gobierno electrónico de Singapur se viene gestando desde la década de 1990. Su estrategia inicial, definida en el año 2000, consistió en desarrollar tres dimensiones: gobierno a gobierno, gobierno a empresa y gobierno a ciudadano. El éxito ha sido rotundo; el 75% de las transacciones con el gobierno han sido realizadas por la vía electrónica. Se han diferenciado el portal a empresas y el portal al ciudadano. Este último tiene la posibilidad de configurar su página personal con los trámites que utiliza.

9 SMS: Short Messaging System. Los mensajes de texto de los teléfonos móviles.

10 Son peajes interestatales. Tienen función de aduanas.

11 Para saber más: <http://www.drishtee.com> , <http://gvandoot.nic.in/>

12 En ingeniería de software, un “entregable” es un producto de un desarrollo.

3.2.3 Marcos conceptuales

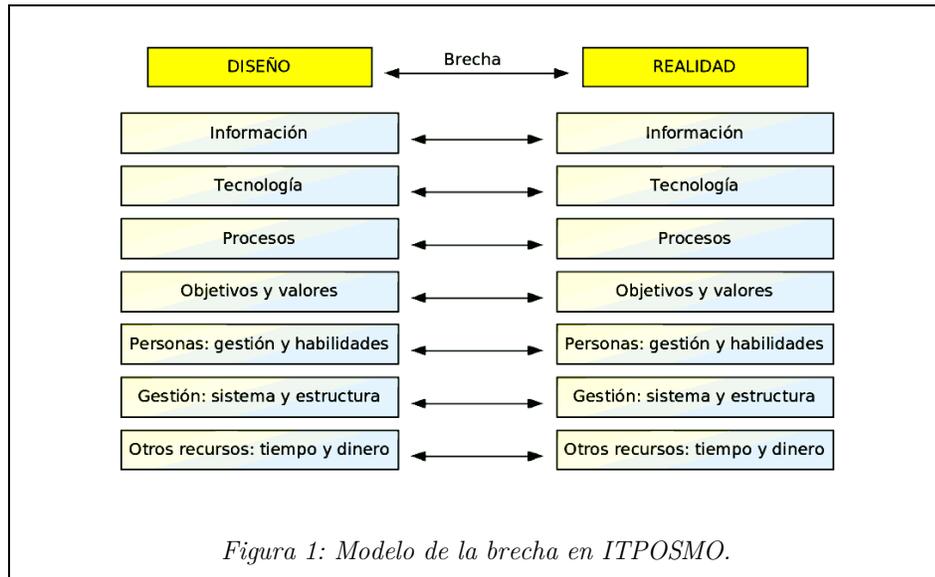
Diferentes modelos conceptuales han sido desarrollados para abordar el tema del gobierno electrónico, ya sea en su totalidad o en aspectos específicos.

ITPOSMO

ITPOSMO es un modelo de análisis propuesto por Richard Heeks [Heeks 2006]. El modelo ITPOSMO fue construido para explicar porqué los desarrollos de gobierno electrónico fracasaban (costaban más de lo que debían, no cumplían plazos, no terminaban). Este modelo considera la observación de las siguientes dimensiones:

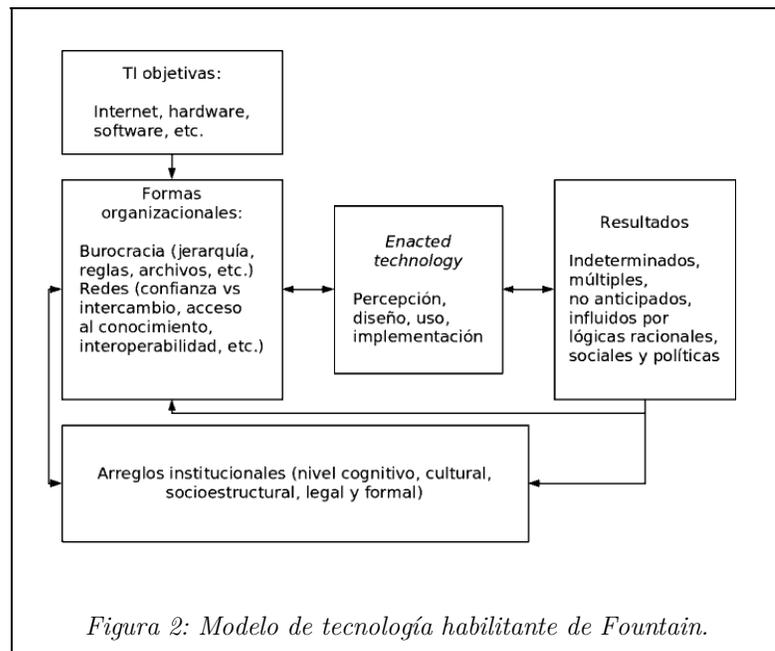
1. Información (Information): considera la información formal del sistema y la información informal manejada por la gente:
2. Tecnología (Technology): considera la tecnología a usar y la disponible, de cualquier tipo.
3. Procesos (Processes): considera las actividades relevantes para los *interesados* (stakeholders).
4. Objetivos y valores (Objectives and values): considera los objetivos de administración pública, los valores organizacionales, la cultura, etc.
5. Personal y competencias (Staffing and skills): considera las políticas, tamaño y competencias del personal.
6. Sistemas y estructuras de gestión (Management Systems and structures): considera los mecanismos para organizar la operación y uso del sistema de gobierno electrónico.
7. Otros recursos (Other resources): principalmente el tiempo y el dinero disponible.

Este modelo se complementa con una observación entre qué factores generan una brecha entre el diseño y la realidad de la implementación del gobierno electrónico, según cada dimensión (ver figura 1).



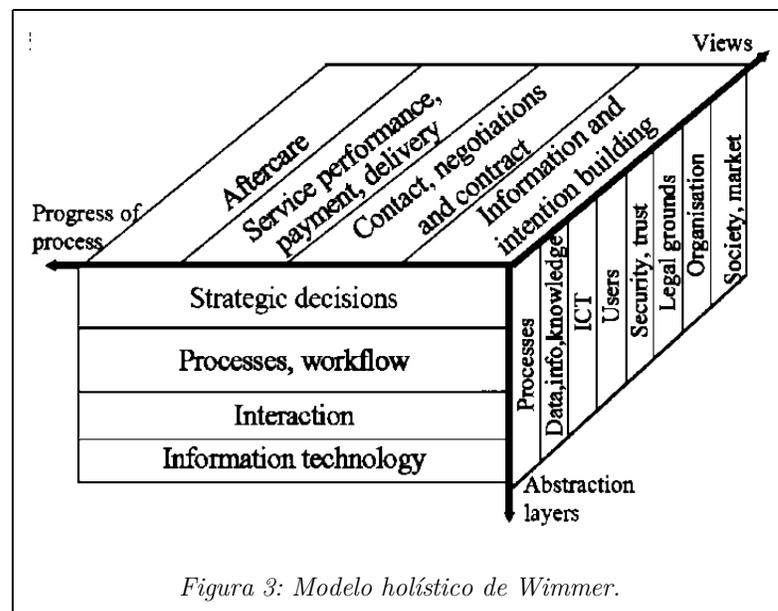
MODELO DE TECNOLOGÍA HABILITANTE

El modelo de tecnología habilitante de Jane Fountain testá orientado a apoyar el diseño de los desarrollos de gobierno electrónico en la asistencia tecnológica [Rivera 2003]. El modelo conceptual se enfoca en la necesidad de hacer útil cada parte del desarrollo. Las flechas de la figura 2 indican qué parte da sentido a qué.



MODELO HOLÍSTICO DE WIMMER

María Wimmer presenta un *modelo holístico* con forma de cubo para el análisis del gobierno electrónico [Rivera 2003], de enfoque *socio-técnico*, para el desarrollo de servicios. Este modelo fue construido, al igual que el modelo ITPOSMO de Heeks, para explicar porqué los proyectos de gobierno electrónico fracasaban (y se creó el concepto de tecnología habilitante). El modelo está pensado para relacionar los niveles de abstracción del desarrollo del gobierno electrónico (niveles de abstracción o *abstraction layers*) con los requerimientos de los interesados (diferentes vistas o *different views*) y con su evolución en el tiempo (progreso de un servicio público o *progress of a public service*). El modelo de cubo se presenta en la figura 3.



3.3 Las tecnologías disponibles¹³

Las tecnologías más relevantes para la implementación del gobierno electrónico son las tecnologías web. Entre éstas destacan las tecnologías XML y los servicios web. Esta sección está basada en la información dispuesta por la W3C, *World Wide Consortium*¹⁴.

3.3.1 El formato documental XML

XML significa *eXtensible Markup Language*, que se puede traducir a lenguaje extensible de

¹³ En el Apéndice A se puede encontrar una versión técnicamente más detallada de este capítulo.

¹⁴ World Wide Consortium, W3C: <http://www.w3.org>

etiquetas (*markups* o *tags*). XML es un formato de texto muy flexible y legible, lo que lo ha hecho muy popular. Algunos ejemplos de archivos populares en formato XML son: XHTML para páginas WEB, SVG -*Scalable Vector Graphics*- para gráficos vectoriales, WML para páginas WAP, BPML para modelación de procesos de negocios, etc.

Para la versión 1.1 de XML (abril de 2004), el equipo de trabajo de XML utilizó los siguientes criterios de diseño:

1. XML deberá ser usado fácilmente en Internet.
2. XML deberá soportar una gran variedad de aplicaciones.
3. XML será compatible con SGML.
4. Deberá ser fácil escribir programas que procesen documentos XML.
5. El número de elementos adicionales a usar en XML se mantendrá en el mínimo absoluto (idealmente cero).
6. Los documentos XML deberán ser legibles por las personas y deberán ser razonablemente claros.
7. El diseño XML deberá ser preparado rápidamente.
8. El diseño XML deberá ser formal y conciso.
9. Los documentos XML deberán ser fáciles de crear.
10. La rigidez de las etiquetas XML será de mínima importancia.

Un documento XML está compuesto de *entidades*. Una entidad puede tener *atributos*, texto o más entidades. El siguiente ejemplo (código 1) muestra una entidad “texto” que contiene el texto “Este es un ejemplo”:

```
<texto> Este es un ejemplo </texto>
```

Código 1: Ejemplo XML sencillo.

Un ejemplo algo más completo, con anidamiento, se puede ver en el código 2.

```
<parrafo>
  <texto>
    Ahora hay dos entidades
  </texto>
  <texto>
    anidadas dentro de otra
  </texto>
</parrafo>
```

Código 2: Dos entidades <texto> dentro de <parrafo>.

Como se ha visto, el formato XML es bastante sencillo y permite representar información a través de su estructura. Además, su sintaxis es clara, lo que facilita su comprensión por parte de los programas, facilitando la compatibilidad entre programas que procesan XML.

Respecto a la capacidad de representar información, dos temas surgen inmediatamente:

1. LA EXISTENCIA DE MODELOS DE DATOS: no es posible relacionar datos de cualquier forma. Y en un cierto contexto, un documento XML no puede asumir cualquier estructura.
2. LA MANIPULACIÓN DE LA INFORMACIÓN: un documento XML tiene relaciones entre datos, esto permite la posibilidad de *consultar* información. Y así mismo, de presentarla de otra manera, o sea, *transformarla*.

ESQUEMA DE UN DOCUMENTO XML

Un documento XML, en un determinado contexto, no puede tener una estructura cualquiera. A esta estructura se le denomina *schema* o *esquema*. Un documento válido sigue su esquema asociado.

Existen varios lenguajes para definir el esquema de un documento XML. Entre esta lista se encuentran:

- **DTD:** DTD significa *Document Type Definition*. DTD tiene la virtud de expresar de forma muy sencilla un esquema. Sin embargo, su poder descriptivo es bastante limitado. Está escrito en XML.
- **RELAX:** RELAX significa *Regular Language description for XML*. Tiene una

estructura y un poder expresivo muy similar al de DTD. Está escrito en XML.

- **TREX:** TREX significa *Tree Regular Expressions for XML*. Su poder expresivo es diferente al de RELAX y DTD, pues describe los anidamientos de entidades a través de anidamientos. Está escrito en XML.
- **RELAX NG:** Es un derivado de RELAX y TREX, combinando su sintaxis.
- **XML Schema:** Poderoso lenguaje para definir el esquema de un documento XML. Es el lenguaje predominante a la hora de definir esquemas de documentos XML.

Un documento XML siempre debería hacer una referencia hacia la definición de su esquema, para verificar su validez.

Además, un documento XML puede incluir varias estructuras diferentes. Debido a esto, existe la distinción de *namespaces* o “espacios de nombres”. Los espacios de nombres evitan los conflictos entre diferentes esquemas dentro de un mismo documento.

MANIPULACIÓN DE DOCUMENTOS XML

World Wide Web Consortium ha definido una multitud de lenguajes para la manipulación de documentos XML. Actualmente existen muchas aplicaciones dedicadas a usar estas tecnologías.

Existen varios lenguajes para consultar y transformar un documento XML. Entre esta lista se encuentran:

- **XPath:** Lenguaje muy sencillo para navegar dentro de un documento XML. Su sintaxis hace honor a su nombre: define “rutas” para acceder a la información. No está escrito en XML. XPath es usado por otros lenguajes, como XQuery y XSLT.
- **XQuery:** Lenguaje de consulta de documentos XML. Extiende XPath, usando mucha de su notación, y permite estructuras de consulta mucho más elaboradas, similares a las consultas SQL, *Standard Query Language*.
- **XSLT:** XSLT significa *XSL Transformations*, donde XSL significa *eXtensible Stylesheet Language*. XSLT es parte de XSL. Actualmente, XSLT es usado de forma intensa en sitios web, transformando documentos XML a HTML adecuados al usuario (según su navegador, idioma, etc.). XSLT está escrito en XML.

3.3.2 Los servicios web

Desde sus primeros días, las tecnologías web dieron espacio a la computación distribuida (documentos HTML invocando scripts CGI). Esta interacción fue creciendo a medida que las tecnologías disponibles fueron mejorando. En el caso específico de las tecnologías XML, estas aplicaciones de computación distribuida son los “servicios web”.

Un servicio web es una funcionalidad disponible en la web. Tanto sus entradas como sus salidas pasan por Internet. Por eso mismo, es posible que un servicio web sea cliente de otro. Y esto se facilita con la utilización de XML. En este sentido, los servicios web gozan de gran potencial para interoperar.

ARQUITECTURA ORIENTADA AL SERVICIO: SOA

Es posible construir grandes redes de servicios web con un objetivo específico. En estas redes, los servicios web interactúan entre sí.

Una arquitectura orientada al servicio es una arquitectura de computación distribuida, en la que sus átomos constituyentes son servicios. Estos servicios operan entre sí, siendo clientes y servidores unos de otros. Así, una aplicación SOA realiza su función al utilizar cada componente.

Los desafíos de una SOA, como los de cualquier arquitectura computacional distribuida, son:

1. Los problemas salidos de la latencia y baja disponibilidad del sistema de transporte de mensajes (la comunicación).
2. La ausencia de un espacio de memoria compartido entre cliente y servidor.
3. Los numerosos problemas introducidos por escenarios de falla parcial.
4. Los desafíos de acceso concurrente a datos remotos.
5. La fragilidad del sistema si se realizan actualizaciones incompatibles a algún participante.

En general, SOA y los servicios web deben utilizarse cuando:

1. Deben operar sobre Internet cuando la disponibilidad y la velocidad no están garantizadas.

2. No hay, o es difícil, mantener un desarrollo organizado de forma centralizada (por ejemplo, cuando es muy difícil realizar actualizaciones simultáneas).
3. Cuando los componentes deben funcionar sobre plataformas y sistemas diferentes.
4. Cuando una aplicación existente necesita ser presentada en una red, y puede ser portada a Internet.

LOS ESTÁNDARES DE LA **W3C** PARA LOS SERVICIOS WEB

Existe una serie de tecnologías asociadas a los servicios web que ya forman parte de la ecología tecnológica de la web.

- **XML**: XML resuelve bastantes retos de interoperabilidad de servicios web. Es por esto que se ha vuelto una tecnología indispensable para éstos.
- **SOAP**: SOAP provee un marco de trabajo estándar, extensible y componible, para empacar e intercambiar mensajes XML, independiente del protocolo de transferencia (SMTP, HTTP, FTP, etc.). SOAP hace alusión a un protocolo de mensajes (comunicación) entre servicios web y a la arquitectura SOA. El protocolo de mensajes está ampliamente soportado por los lenguajes de programación.
- **WSDL**: WSDL significa *Web Services Description Language*. Este lenguaje detalla el funcionamiento del servicio web, a nivel de mensajes y funcionalidades asociadas. Está escrito en XML.
- **WS-CDL**: significa *Web Services Choreography Description Language*. WS-CDL indica cómo se relacionan los servicios web para interoperar. Indica qué servicios web se comunican entre sí, entre otras cosas. Está escrito en XML.

3.4 La evaluación socio-económica¹⁵

La evaluación socio-económica es un tipo de evaluación de proyectos que considera el valor que entrega un proyecto a la sociedad en su totalidad. Ello contrasta con la evaluación privada de proyectos, que sólo considera el beneficio que un proyecto reporta a los inversionistas.

La evaluación socio-económica de proyectos es utilizada en el contexto de la administración pública. El objetivo de la evaluación consiste en determinar el beneficio que un proyecto entrega

¹⁵ Esta sección está basada en [Contreras 2004].

al país. En este sentido, la evaluación socio-económica es un elemento indispensable para la toma de decisiones en el gobierno.

3.4.1 Criterios de aprobación de proyectos y políticas

Desde una perspectiva económica, el interés del gobierno es aumentar el *bienestar* de toda la sociedad. La aprobación de proyectos adecuados es de mucha importancia.

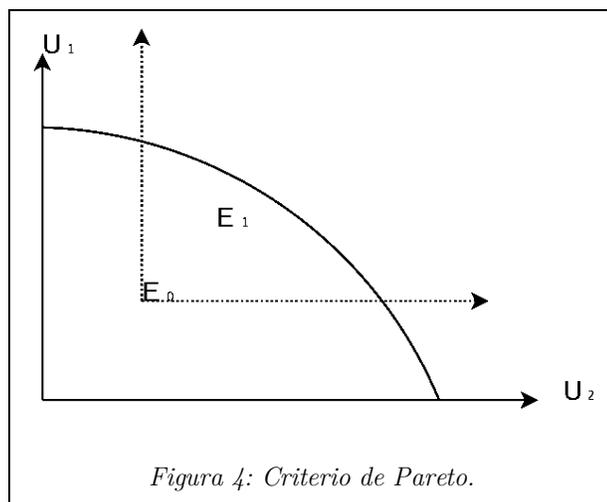
A continuación se presenta una breve selección de criterios orientados a la distribución eficiente de recursos.

CRITERIO DE PARETO

El criterio de Pareto para la aprobación de proyectos exige que *al menos un individuo se vea beneficiado por el proyecto y que nadie se vea perjudicado*. En términos de funciones de utilidad, el criterio de Pareto exige:

$$\sum_k U_k^1 > \sum_k U_k^0, \quad U_k^1 \geq U_k^0, \quad \forall k$$

En la fórmula anterior, U_k^p representa la utilidad del individuo k en el estado p , donde p es 0 en la situación sin proyecto y 1 en la situación con proyecto.



El gráfico anterior muestra, en una economía de dos individuos, la aprobación de un proyecto según el criterio de Pareto. Nótese que E_1 es mejor que E_0 , tanto para el individuo 1 como para el individuo 2. Cualquier estado de la economía ubicado dentro del cuadrante definido

por las líneas punteadas es aceptable según el criterio de Pareto, y se le dice que E_1 es Pareto-superior a E_0 . Si E_1 se ubica en la línea curva, la *frontera de posibilidades de la utilidad*, se dirá que es Pareto-eficiente. No es posible seguir mejorando la economía luego de un estado Pareto-eficiente de la economía, sin expandir la frontera.

Un problema importante de este criterio es su dificultad para medir un estado de avance -el crecimiento- de la economía debido a un proyecto.

CRITERIO DE KALDOR-HICKS

El criterio de Pareto es bastante exigente debido a que considera que los individuos no son comparables. Sin embargo, siguiendo esta restricción, existe otro criterio que lidia mejor con esta dificultad: es el criterio de compensación de Kaldor-Hicks.

El criterio de compensación indica que, si bien un estado de la economía puede no ser Pareto-superior a otro, sí es posible redistribuir parte de los excedentes de los individuos beneficiados a los individuos perjudicados. Si el proyecto, luego de esta compensación, es Pareto-superior, entonces es aceptable.

En resumen, si el proyecto genera más beneficios que perjuicios, entonces es aceptable según el criterio de Kaldor-Hicks.

Este criterio sufre de algunos problemas:

- En la práctica, hay perjuicios que no pueden ser compensados por los beneficios (todos aquellos daños irreversibles).
- No siempre ocurre una compensación a los perjudicados tras un proyecto. Y si ocurre, no siempre es adecuada.
- Este criterio no es transitivo. Genera contradicciones al comparar puntos Pareto-óptimos.

CRITERIO DE BERGSON-SAMUELSON¹⁶

Este criterio consiste en la comparación de estados de la economía de acuerdo a una función de bienestar social. Si un proyecto genera un cambio positivo en la función de bienestar social, entonces es aceptable.

16 Este punto está basado en: <http://cepa.newschool.edu/het/essays/paretian/paretosocial.htm>

Una función de bienestar social se expresa de la siguiente forma:

$$W = W(U_1, U_2, \dots) \in \mathbb{R}$$

Básicamente, si un proyecto genera un $\Delta W > 0$, entonces es aceptable.

Este criterio goza de la flexibilidad de la elección de una función de utilidad. Dependiendo de su forma, distintos proyectos pueden ser aceptados.

OTROS CRITERIOS

Existen muchos otros criterios para decidir proyectos. A continuación se presentan algunos:

CRITERIO DE SCITOVSKY: Sigue la línea del criterio de Pareto. Este criterio se resume en: “Si, según Kaldor-Hicks, E_0 es preferible a E_1 , pero el inverso no es cierto, entonces E_0 es preferible a E_1 según Scitovsky”. Este criterio sigue teniendo problemas de completitud.

CRITERIO PATERNALISTA O DICTATORIAL: No es realmente un criterio de decisión, sino una forma de tomar decisiones. Tiene varias objeciones éticas.

CRITERIO SEGÚN VOTACIÓN: Este criterio se usa bastante actualmente. Sin embargo, no asegura correctitud. Si se decide entre tres o más opciones, puede no ser satisfactorio (paradoja de Condorcet).

Los dos últimos criterios no son excluyentes respecto de los criterios anteriores.

3.4.2 La teoría de la evaluación social de proyectos

La evaluación socio-económica de proyectos sigue la línea del criterio de bienestar social, en el que existe una función de bienestar social. Básicamente, se puede decir que la función de bienestar social es función del bienestar de sus individuos, y que el bienestar de un individuo depende de su consumo. En otras palabras, la función de bienestar social se puede expresar como una función del consumo (riqueza) de cada individuo.

Suponiendo que un proyecto genera un cambio marginal sobre la sociedad, se puede hacer la siguiente aproximación de primer orden¹⁷:

17 En [Contreras 2004] se hace una prueba con más significado económico, evitando la aproximación de primer orden. Sin embargo, el memorista ha observado que utilizar la aproximación simplifica mucho el proceso deductivo y no le resta validez.

$$\Delta W \approx \sum_k \left(\frac{\partial W}{\partial C_k} \right) \Delta C_k$$

En la ecuación anterior, C_k representa el consumo o riqueza del individuo k . Simplificando la notación:

$$\Delta W \approx \sum_k \phi_k VP_k$$

En la fórmula anterior, ϕ es un ponderador por individuo y VP es el valor del proyecto para un individuo (es el *valor presente*, que se verá más adelante).

A continuación se verá un par de enfoques o *escuelas* de evaluación social de proyectos, desde la perspectiva de la función de bienestar social.

ENFOQUE DE EFICIENCIA

Este enfoque opera con los siguientes principios:

1. El beneficio de una unidad adicional de un bien o servicio para un comprador es medido por su precio de demanda.
2. El costo de oportunidad de una unidad adicional de un bien o servicio para un proveedor es medido por su precio de oferta.
3. Un peso para una persona vale lo mismo que un peso para otra persona (todas las personas son iguales).

En este caso, ϕ es igual para todos los individuos. Luego:

$$\Delta W \approx \phi \sum_k VP_k = \phi VP$$

Nótese que si la suma de los valores presentes es positiva, entonces el proyecto es aceptable. Además, nótese que este método de evaluación es equivalente al criterio de Kaldor-Hicks; si los beneficios son superiores a los perjuicios, entonces el proyecto es aceptable.

ENFOQUE DISTRIBUTIVO

El enfoque de eficiencia busca que la riqueza del país aumente. Sin embargo, no lidia con la

distribución de la riqueza. En contraste, el enfoque distributivo busca aprobar proyectos que consideren una distribución más equitativa de la riqueza.

En el enfoque distributivo, los ponderadores ϕ_k se utilizan para dar más importancia a los más pobres. Sin embargo, es importante notar que la función de bienestar social no es *observable*. En este sentido, los ponderadores ϕ_k se eligen de manera conveniente.

Un ejemplo de aproximación práctica para ϕ_k es el siguiente:

$$\phi_k = \left(\frac{Y}{Y_k} \right)^\alpha$$

En la fórmula anterior, k representa un quintil de ingreso, Y_k representa el ingreso per cápita de ese quintil, Y representa el ingreso per cápita promedio del país y α es un parámetro de sensibilización ($\alpha \in [0,1]$).

Esta fórmula es muy conveniente. Si el parámetro de sensibilización es cero, se tiene el enfoque de eficiencia. Si, por el contrario, es uno, se tiene un gran énfasis en la redistribución. Además, implica el cálculo del proyecto por quintil de ingreso, agrupando cómodamente los resultados.

3.4.3 La práctica de la evaluación social

El enfoque de eficiencia es el enfoque predominante de evaluación socio-económica en Chile. De todas maneras, es posible usar el enfoque distributivo utilizando ponderadores y evaluación por segmento.

EL VALOR PRESENTE NETO

El valor presente, o valor actual, es el valor que un flujo de dinero en un momento cualquiera posee en este momento. Es distinto recibir mil pesos hoy que recibir mil pesos en un mes más. El valor presente lidia con el efecto del tiempo en los flujos de dinero.

¿Por qué el tiempo afecta el valor del dinero? Por el costo de oportunidad de éste. Si una persona tiene \$1000 ahora, los puede invertir y tener \$1100 el mes que viene. Tener \$1000 el mes que viene significa no poder invertir el dinero por una cantidad de tiempo. El costo de oportunidad de \$1000 son \$100 para esta persona.

¿Cómo es la forma del costo de oportunidad? Su forma es la de una *tasa*. Por ejemplo, supongamos que una persona tiene \$1000 y los convierte en \$1100 durante un mes. Tener \$2000 significa tener dos veces \$1000. En este sentido, haría el mismo trabajo con cada lote de \$1000, obteniendo \$1100 con cada uno. El resultado, es que los \$2000 se convirtieron en \$2200. El costo de oportunidad es proporcional a la cantidad de dinero. Con \$1000, se obtuvieron \$100 extra. Con \$2000, se obtuvieron \$200. Esta persona tiene un costo de oportunidad del 10% en un mes. Si tuviera \$3000, obtendría \$300 extra.

Suponiendo que el costo de oportunidad no cambia, si una persona tiene un costo de oportunidad del 10% mensual simple, y tiene \$1000 hoy, tendrá \$1100 en un mes más. Esos \$1100 volverán a ser usados y la persona tendrá \$1210 el mes siguiente. Y así sucesivamente. Si r es la tasa de costo de oportunidad, el dinero en el mes t será: $D(t) = \$1000 \times (1+r)^t$

¿Cómo comparar flujos de dinero en momentos diferentes? Con un costo de oportunidad del 10% mensual, \$1000 hoy valen lo mismo que \$1100 el próximo mes (si su destino es la inversión), pues ambos valen \$1100 el próximo mes. Pero esto significa que tener \$1100 el próximo mes es equivalente a tener \$1000 ahora.

Los ejemplos anteriores sirven para ilustrar la forma general del *valor presente* para un flujo de dinero (r es el costo de oportunidad):

$$VP = \frac{\text{Dinero}_t}{(1+r)^t}$$

Un proyecto origina flujos durante diferentes períodos de tiempo. Por ello, el valor actual de un proyecto consiste en la suma de todos los flujos que origina. La riqueza total originada por un proyecto es calculada mediante el **valor presente neto**. Su fórmula, para un costo de oportunidad r constante, es:

$$VPN = \sum_{t \geq 0} \frac{F_t}{(1+r)^t}$$

Se ha denotado F_t a un flujo en un momento t (generalmente medido en años). Nótese que este es un flujo de valor. No necesariamente tiene que ser de dinero. Si ocurre un flujo de mercancías, el flujo corresponderá al valor en dinero de esas mercancías (generalmente).

Una notación más conveniente consiste en identificar la inversión. La inversión de un proyecto es un costo inicial causado por la implementación del proyecto. La fórmula del valor

presente neto, tomando a I como la inversión, es:

$$VPN = -I + \sum_{t \geq 1} \frac{F_t}{(1+r)^t}$$

A r también se le denomina “tasa de descuento”. En el caso de evaluaciones socio-económicas, se le denomina “tasa social de descuento”.

LAS DISTORSIONES DEL MERCADO

El enfoque de eficiencia tiene problemas cuando los precios de mercado se encuentran sometidos a “distorsiones”. Estas distorsiones son beneficios y perjuicios no medidos a través del mercado. Éstas son las externalidades positivas y negativas, respectivamente.

La evaluación social, al considerar a todos los individuos de la sociedad, debe lidiar con estos problemas. En la práctica, el problema se corrige a través del uso de *precios sociales*. Los precios sociales existen para reemplazar los precios de mercado en la evaluación socio-económica. Son provistos anualmente por el MIDEPLAN. La tasa social de descuento también es publicada en este documento.

3.4.4 Indicadores de evaluación

Los indicadores de evaluación de proyectos son útiles para decidir entre distintos proyectos o cuando existen otros elementos en la evaluación: interferencias entre proyectos, restricción de presupuesto, complicación en las estimaciones, etc. En el caso sencillo, sólo el cálculo del VPN es necesario. Pero en casos más complicados, se hace necesario usar otros criterios.

IVAN¹⁸ (CRITERIO)

Cuando existen restricciones de presupuesto, y se desean hacer varios proyectos, el criterio del IVAN es muy conveniente como primera aproximación.

El problema es el siguiente: existen varias propuestas de proyectos a realizar, pero existe una restricción de presupuesto que no permite realizar todos los proyectos. Supóngase que son conocidos I_k, VPN_k para cada proyecto k . Y si la restricción de capital es R , el criterio del IVAN es como sigue:

18 Este criterio no está presente en [Contreras 2004]. Sin embargo, su presentación es tradicional.

1. Calcular el IVAN para cada proyecto: $IVAN_k = \frac{VPN_k}{I_k}$
2. Ordenar los proyectos según IVAN, de mayor a menor.
3. Agregar los proyectos a la cartera de realización, considerando que la suma de las inversiones tiene que ser menor que R .

El beneficio total según este método corresponde a la suma de los VPN.

Este problema es un caso particular del *problema de la mochila*¹⁹, un problema clásico de complejidad computacional. El problema de la mochila, también llamado el problema del ladrón, consiste en colocar objetos de valor en una mochila, buscando obtener el máximo valor total posible considerando que la mochila tiene espacio limitado y los objetos son de diferentes tamaños. Hay otra versión, equivalente, en que la restricción ocurre sobre el peso a cargar. El algoritmo goloso presentado obtiene una muy buena aproximación al óptimo.

El criterio del IVAN tiene problemas cuando hay interferencias en los proyectos. Hay casos en que realizar un proyecto reduce el valor presente de otro proyecto. Hay casos en que los proyectos son excluyentes entre sí. Hay casos en que hay otras restricciones además del presupuesto. Además, los proyectos pueden interferir a futuro en términos presupuestarios. Todo este tipo de interacciones reduce la calidad del criterio del IVAN. En estos casos, conviene realizar un programa matemático para resolver estos problemas.

INDICADORES COSTO-BENEFICIO

El IVAN es un caso especial, de uso más preciso, de este tipo de indicadores. Los indicadores costo-beneficio o razones costo-beneficio, son razones de la forma:

$$\text{Indicador}_k = \frac{\text{Beneficio}_k}{\text{Costo}_k}, k \in \text{Proyectos}$$

Este tipo de indicadores pretende revelar cuánto se aprovecha cada peso de costo en términos de beneficios. En otras palabras, es una medida de la *rentabilidad* de una inversión.

De nuevo considerando recursos limitados, este tipo de indicadores sirve para sacar el mejor

¹⁹ Esta es una observación del memorista, basado en sus conocimientos de ciencias de la computación.

provecho -aproximadamente²⁰- de los recursos invertidos. Nótese que este tipo de indicadores sólo sirve para comparar proyectos.

INDICADORES COSTO-IMPACTO

Cuando la medición de los beneficios es difícil de realizar, por calidad de la información, observabilidad, razones éticas u otros motivos, se hace necesario comparar los proyectos de una forma heurística²¹. Los costos suelen ser fáciles de medir, así que se dejan en pesos. En vez del beneficio, se usa alguna unidad del impacto del proyecto.

Una razón costo-impacto asume la forma:

$$\text{Indicador}_k = \frac{\text{Impacto}_k}{\text{Costo}_k}, k \in \text{Proyectos}$$

Por ejemplo, sea el caso de los semáforos. Por impacto, podría tomarse el número esperado de personas que salva cierto tipo de semáforo. El costo es el precio del semáforo (lo usual) más el costo de instalarlo. La unidad del indicador quedaría en [vidas/pesos] o [personas/pesos]. Si se dispone de presupuesto limitado, este indicador se hace muy útil para determinar qué tipo de semáforos instalar. Además, se evita la complicación moral de evaluar económicamente la vida humana.

3.5 Las métricas en la ingeniería de software

Una métrica consiste básicamente en la medición numérica de un atributo. Existen debido a la necesidad de tener información cuantitativa para la toma de decisiones.

3.5.1 Problemas de las métricas

Las métricas para la ingeniería de software tienen poco más de 30 años de historia, y su utilización actual en los proyectos de software es bastante baja. Uno de sus grandes problemas es desestimar la validez de las mediciones.

²⁰ Nuevamente, se presenta el problema de la mochila. Por ende, el uso de este tipo de indicadores sólo entrega resultados aproximados al óptimo.

²¹ Algo parecido al problema de la mochila, sólo que ahora los indicadores son aproximaciones a los indicadores costo-beneficio. La dificultad de medir el impacto de una manera precisa hace menos fiable al criterio de decisión. Por ello, tampoco se puede esperar mucho más.

Una de las principales dificultades proviene de la medición de atributos difíciles de medir debido a su subjetividad. En la figura 5 (página 28), obtenida de [Fenton 2000], se ilustra de forma clara este problema.

Los atributos subjetivos como “calidad”, “mantenibilidad” o “usabilidad” son difíciles de medir sin un contexto bien definido. ¿Qué es el “tamaño” de un desarrollo de software? Este es un problema interesante. ¿Qué es, precisamente, la “reusabilidad”? Distintos equipos de desarrollo de software pueden reutilizar componentes de manera distinta.

ENTIDADES	ATRIBUTOS	
	<i>Internos</i>	<i>Externos</i>
Productos		
Especificaciones	Tamaño, reuso, modularidad, redundancia, funcionalidad...	Comprensibilidad, mantenibilidad...
Diseños	Tamaño, reuso, modularidad, acoplamiento, herencia...	Calidad, complejidad, mantenibilidad...
Código	Funcionalidad, complejidad algorítmica, control de flujo...	Disponibilidad, usabilidad, reusabilidad...
Datos de prueba	Tamaño, nivel de cubrimiento...	Calidad, reusabilidad...
...		
Procesos		
Construir	Tiempo, esfuerzo, número de cambios de requerimientos...	Calidad, costo, estabilidad...
Diseño detallado	Tiempo, esfuerzo, número de faltas de especificación...	Costo, costo-efectividad...
Pruebas	Tiempo, esfuerzo, número de faltas de código...	Costo, costo-efectividad, impacto...
...		
Recursos		
Personal	Edad, salario...	Productividad, experiencia, inteligencia...
Equipos	Tamaño, nivel de comunicación, estructuración...	Productividad, calidad...
Organizaciones	Tamaño, certificación ISO, nivel CMM	Madurez, rentabilidad...
Software	Precio, tamaño...	Usabilidad, disponibilidad...
Hardware	Precio, velocidad, tamaño de la memoria...	Disponibilidad...
Oficinas	Tamaño, temperatura, iluminación...	Comodidad, calidad...
...		

Figura 5: Atributos y métricas [Fenton 2000].

La otra parte del problema corresponde a los atributos medidos. Su validez como representantes de los atributos subjetivos puede ser bastante discutible. Por ejemplo, se suele utilizar el “número de líneas de código” (LOC) como métrica del tamaño de un software. ¿Qué es una línea de código? ¿Cuánto hace una línea de código? Estos son problemas de esta métrica, ya que una línea de código no tiene un significado fijo. Además, distintos programadores pueden escribir distintas líneas de código según sus estilos de programación y su estrategia de implementación.

[Kaner 2004] presenta, a modo de ejemplo, una breve lista de métricas de este tipo. A continuación se presentan y se discuten algunas.

- Tamaño del código fuente (LOC, KLOC). Esta métrica es peligrosa porque no todo el código fuente es igual de funcional o importante.
- Tiempo de programación (horas-hombre $[H - H]$, meses-hombre).
- Productividad del programador $\left(\frac{\text{tamaño del código}}{\text{Tiempo de programación}} \left[\frac{LOC}{H-H} \right] \right)$. Esta métrica es peligrosa. Si se usa, genera un incentivo en los programadores que los lleva a dilatar el código. Además, no todo el código es igualmente complejo o esencial.
- Densidad de defectos $\left(\frac{\text{Número de fallas}}{\text{Tamaño del código}} \left[\frac{Fallas}{LOC} \right] \right)$. Esta métrica es complicada porque no todo el código es igualmente complejo y no todas las fallas son iguales.
- Estabilidad de requerimientos $\left(\frac{\text{Número inicial de requerimientos}}{\text{Número final de requerimientos}} \right)$. Esta métrica esconde la posibilidad de que los requerimientos hayan sido cambiados completamente.
- Esfuerzo de reparación de fallas (horas-hombre $[H - H]$). Esta es una métrica natural. El problema es que no distingue entre tipos de fallas (hay fallas más difíciles de reparar y ubicar que otras).
- Esfuerzo total del proyecto (horas-hombre $[H - H]$).
- Arruinamiento del sistema $\left(\frac{\text{Esfuerzo destinado a reparar fallas}}{\text{Esfuerzo total del proyecto}} \right)$. Tiene el mismo problema del esfuerzo de reparación de fallas.

3.5.2 Métricas en casos específicos

De todas maneras, el uso de métricas es una necesidad en la ingeniería de software, especialmente cuando se necesita saber el estado de avance de un desarrollo. En este sentido, existen casos específicos de métricas diseñadas para situaciones especiales.

MÉTRICAS DE MÓDULOS PREFABRICADOS (COTS)

Los módulos prefabricados o “*commercial off-the-shelf*” son elementos importantes dentro de un desarrollo a gran escala. Sin embargo, dado que su implementación no se altera -en un principio- durante el desarrollo, se hace necesario conocer más detalles de estos módulos.

[Sahra 2001] presenta una tabla con métricas de nivel de sistema (figura 6) para componentes prefabricados.

Categoría	Métrica	Mide o evalúa
Gestión	Costo	<i>Gasto total del desarrollo de software incluyendo costos de adquisición, integración, mejora, etc.</i>
	Tiempo al mercado	<i>Tiempo que ha pasado desde que se inició el desarrollo hasta que el producto llegó al mercado.</i>
	Ambiente	<i>Capacidad y madurez del ambiente de trabajo en el cual se ha desarrollado el producto.</i>
	Utilización de recursos	<i>Uso de recursos computacionales como una fracción de la capacidad total.</i>
Requerimientos	Conformancia	<i>Adherencia del producto integrado a los requerimientos definidos por fase de desarrollo.</i>
	Estabilidad	<i>Nivel de cambios de los requerimientos definidos para el desarrollo.</i>
Calidad	Adaptabilidad	<i>Capacidad del sistema integrado para adecuarse a cambios de requerimientos.</i>
	Complejidad de interfaces e integración	<i>Interface de componentes y middleware o la complejidad del código integrado.</i>
	Prueba de integración	<i>Fracción del sistema que ha pasado de forma satisfactoria las pruebas de integración.</i>
	Prueba de totalidad	<i>Fracción de la funcionalidad del sistema que ha pasado una prueba de principio a fin.</i>
	Perfiles de error	<i>Número acumulativo de fallas encontradas.</i>
	Disponibilidad	<i>Probabilidad de que el sistema funcione sin fallas durante un período específico de tiempo.</i>
	Satisfacción del consumidor	<i>Grado en el que el software satisface los requerimientos del consumidor.</i>

Figura 6: Métricas de nivel de sistema para COTS [Sahra 2001].

Estas métricas sufren los mismos problemas de las métricas tradicionales.

MODELO COCOMO

El modelo COCOMO es el primer gran modelo de estimación basado en métricas e información estadística [Boehm 1981]. Utiliza estimaciones bayesianas para la estimación del esfuerzo de desarrollo (horas-hombre) según la dimensión del problema, medido según la técnica de los “puntos de función” o miles de líneas de código (KLOC). El uso de los puntos de función (que se pueden estimar según los casos de uso) permiten evitar en la tan poco confiable métrica KLOC. Cabe decir que COCOMO es uno de los métodos de estimación de proyectos de software más usados en la historia de la ingeniería del software.

CUADRO DE MANDO INTEGRAL²²

El Cuadro de Mando Integral (CMI), o *Balanced Scorecard*, es una técnica de control de gestión muy famosa que ha tenido muy buenos resultados. Como se notará, es una técnica que proviene de la gestión privada. Sus creadores son Norton y Kaplan, dos “gurús” del *management*.

El imaginario mental detrás del CMI consiste en un gerente que está viendo un panel de control, donde cada instrumento indica qué está pasando en la empresa. Es análogo al caso de un piloto y el panel de control de un avión: el piloto se guía por las lecturas de los instrumentos para pilotar. Así mismo, el CMI debe asistir al gerente en su gestión.

El CMI consta de un conjunto de indicadores que muestran información tanto del rendimiento del área de mando del gerente (sea la empresa, un departamento, una sucursal, etc.) como del cumplimiento de las metas. Estos indicadores o métricas son algunos muy precisos (aquellos basados en estimaciones económicas, financieras o contables) y otros lejos de ser tales (escalas graduadas de 1 a 5, etc.). Pese al importante número de métricas imprecisas, el CMI ha probado dar muy buenos resultados²³.

El proceso del CMI es el siguiente:

1. Traducir la visión a metas operacionales (tangibles).
2. Comunicar la visión y relacionarla al rendimiento de los empleados.
3. Hacer el plan de negocio.

²² Basado en [Kaplan 1992].

²³ Esto podría llevar a la conclusión que saber algo de forma imprecisa es mejor que no saber nada. Por supuesto, el gerente está al tanto de la imprecisión de esos indicadores, por lo que no hará algo imprudente. De hecho, es muy posible que él mismo haya diseñado las métricas.

4. Retroalimentarse y aprender para ajustar la estrategia.

METODOLOGÍA GOAL-QUESTION-METRIC (GQM)²⁴

La metodología GQM consiste en el seguimiento de objetivos de acuerdo a métricas. Este método está inspirado en el *Control de Calidad Total* (TQM, *Total Quality Management*), una estrategia industrial para el control de calidad. Es posible observar su similitud con el Cuadro de Mando Integral.

La metodología GQM presenta un procedimiento para la elección de métricas. A saber, el procedimiento es el siguiente:

1. Formulación de un objetivo o meta. Este objetivo debe estar enfocado sobre un elemento del desarrollo. Puede ser de:
 1. Productos: Artefactos, entregables, etc.
 2. Procesos: Diseño, implementación, pruebas, etc.
 3. Recursos: Personal, hardware, espacio, etc.
2. Formulación de una pregunta. Esta pregunta debe relacionar información sobre el elemento de desarrollo a la meta.
3. Formulación de métricas. Deben utilizar los datos necesarios para responder la pregunta. Los datos pueden ser:
 1. Objetivos: Dependen del objeto y no del observador. Ej. tiempo invertido.
 2. Subjetivos: Dependen del observador. Ej. legibilidad de un texto.

La metodología se debe seguirse para cada objetivo. Como una observación, una métrica puede satisfacer preguntas de más de un objetivo.

Otra observación: La calidad de las métricas es responsabilidad exclusiva de quien las define. Este modelo no lidia con la validez de éstas.

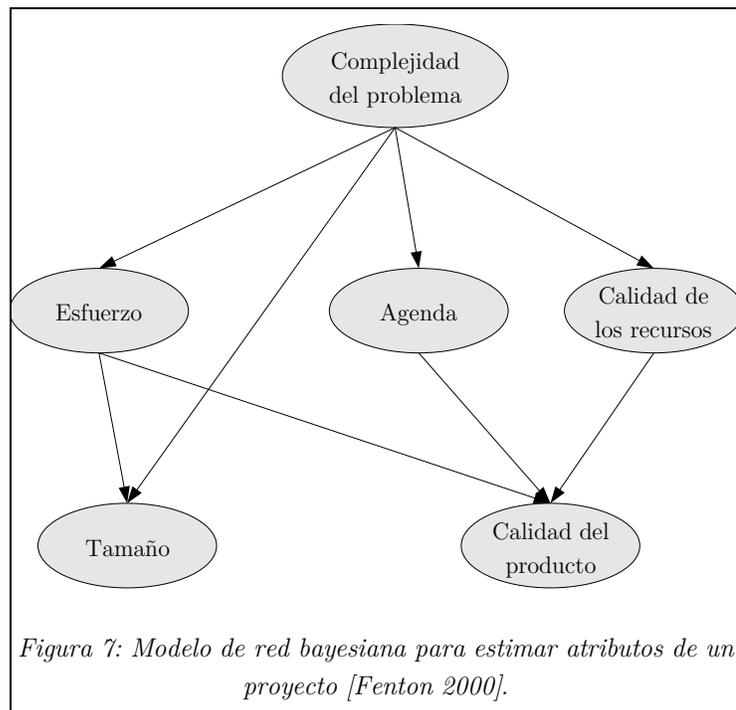
²⁴ Esta sección está basada en [Basili 2002].

MÉTRICAS BAYESIANAS²⁵

Las métricas bayesianas están basadas en redes bayesianas o redes de inferencia. Estas redes se van alimentando de la experiencia de sus usuarios y se van ajustando para entregar resultados esperados. El problema de estas métricas es que se ajustan al punto de vista de la persona que las alimenta.

Las redes bayesianas sólo aceptan parámetros lógicos. Por eso, es necesario definir intervalos para ingresar información numérica a una red bayesiana. Lo mismo vale para sus salidas.

Por ejemplo, en [Fenton 2000] se presenta un esquema general²⁶ de un modelo causal para estimar características generales de un proyecto de software.



²⁵ Esta sección está basada en [Fenton 2000].

²⁶ No viene al caso, pero este modelo se puede simplificar bastante debido a que todo es función de la complejidad del problema. Seguramente, el modelo tiene otros *parámetros* que justifican su forma.

3.5.3 Definiciones formales²⁷

Un nuevo formalismo ha surgido debido a la ambigüedad de las métricas en la ingeniería del software.

Definición. *Se define **medición** como el mapeo del mundo empírico al mundo formal, relacional.*

Definición. *Una **medida** es el número o símbolo asignado a una entidad por un mapeo (medición) de manera de caracterizar un atributo.*

El estándar IEEE 1061 define *atributo, factor de calidad, métrica y métrica de calidad de software.*

Definición. *Un **atributo** es una propiedad medible física o abstracta de una entidad.*

Definición. *Un **factor de calidad** es un atributo de software orientado a la gestión que contribuye a su calidad.*

Definición. *Una **métrica** es una función de medición (un mapeo).*

Definición. *Una **métrica de calidad de software** es una función cuyas entradas son datos de software y su salida es un valor numérico simple, que se puede interpretar como el grado en que el software posee un atributo que afecta su calidad.*

La validez de una métrica es un asunto importante. En este sentido, una “métrica directa” representa la medición más esencial de un atributo.

Definición. *Una **métrica directa** es una métrica que no depende de la medida de ningún otro atributo.*

Definición. *Una **métrica indirecta** o **derivada** es una métrica que depende de la medida de otro atributo. Es la métrica que no es directa.*

Una métrica directa, debido a su simplicidad, se asume por válida (correcta). Esto contrasta a la métrica indirecta -cuyo dominio es una *n-tupla*-.

El estándar 1061 presenta un criterio de validación de métricas:

²⁷ Esta sección está basada en [Kaner 2004].

1. Correlación. La métrica debe estar linealmente relacionada al factor de calidad que mide (correlación estadística).
2. Consistencia. Si F es un atributo de calidad y M es una función métrica, $M: F \rightarrow Y$ debe ser monótona, tal que: $f_1 < f_2 < f_3 \Rightarrow y_1 < y_2 < y_3$.
3. Seguimiento. Para $M: F \rightarrow Y$, si f cambia de f a f' en un período de tiempo, $M(f)$ debe reflejar propiamente ese cambio (y a y').
4. Predictividad. Para $M: F \rightarrow Y$, si se conoce el valor de Y en algún instante, debería ser posible predecir el valor de F en ese instante.
5. Poder discriminador. Una métrica de calidad debiera distinguir entre componentes de software de alta calidad y baja calidad. Ello debiera reflejarse de forma clara numéricamente.
6. Disponibilidad. Una métrica deberá demostrar correlación, seguimiento, consistencia, predictividad y poder discriminador, para al menos un P% de la aplicación de la métrica.

[Kaner 2004] discute este criterio de validación de métricas debido a que mantiene la ambigüedad del atributo a medir. Por esto, propone el siguiente criterio de evaluación a aplicar a una métrica:

1. **¿Cuál es el propósito de la medida?** ¿Es relevante?
2. **¿Cuál es el alcance de la medida?** ¿Se mide por una persona? ¿Necesita un año de estadísticas? ¿Necesita información de toda la compañía?
3. **¿Qué atributos se tratan de medir?** Se necesita una idea precisa. La métrica pierde credibilidad ante una idea vaga.
4. **¿Cuál es la escala natural del atributo que se desea medir?** Esto se hace más complicado en ciertos casos, como el tamaño de programa.
5. **¿Cuál es la variabilidad natural de cada atributo?** El atributo puede cambiar en el tiempo. Además, su medición puede tener errores de precisión naturales (por ejemplo, imprecisión humana).
6. **¿Cuál es la métrica?** (la función) ¿Cómo se mide? ¿Con qué instrumento?

Muchas métricas sólo consisten en contar, otras requieren comparaciones o clasificaciones, otras requieren medir el tiempo, etc.

7. **¿Cuál es la escala natural de la métrica?** Si no tiene escala, podría ser ordinal (mapea una relación de orden), cardinal, etc.

8. **¿Cuál es la variabilidad natural de las lecturas del instrumento?** Este es el error de medición.

9. **¿Cuál es la relación entre el atributo y el valor de la métrica?** Esta pregunta sirve para validar la construcción de la métrica. Sirve para observar la presencia de un modelo y su validez en esta situación.

10. **¿Cuáles son los efectos naturales y predecibles de usar este instrumento?** ¿Se puede manipular el resultado de la métrica? Por ejemplo, un incentivo para encontrar errores puede alterar el número de errores encontrados por el equipo de pruebas.

Como se puede apreciar, la evaluación de la validez de las métricas es un tema complicado. Por eso, actualmente sólo existen recomendaciones. De todas maneras, la ciencia formal establece modelos matemáticos para determinar métricas, o bien, las métricas salen de fenómenos físicos repetibles (como el alargamiento de un resorte, mediciones con reglas, etc), donde los efectos subjetivos son poco influyentes.

3.6 Resumen

En este capítulo se hizo una breve revisión del estado del arte del gobierno electrónico a nivel internacional, las tecnologías disponibles, la evaluación socio-económica y la teoría de las métricas.

En la sección del gobierno electrónico, se vieron los conceptos de:

- La Nueva Gestión Pública, una corriente de administración pública basada en principios y prácticas de la gestión privada.
- La Gobernanza Pública, una corriente de administración pública basada en la estructura institucional y su modificación según las redes de influencia.
- La Sociedad de la Información, como un nuevo esquema social basado en las tecnologías de la información y las comunicaciones.

- La Nueva Economía, nacida de la economía de Internet y las redes.

También se vieron algunos marcos de análisis del gobierno electrónico:

- ITPOSMO y el Modelo de la Brecha, ambos de Heeks.
- Las Tecnologías Habilitantes de Fountain.
- El Modelo Holístico de Wimmer.

En las tecnologías disponibles, se dio énfasis a las tecnologías web, que se dividen en las tecnologías XML y las tecnologías de servicios web. En cuanto a XML, se hizo una breve revisión del formato XML, los lenguajes de definición de esquemas y los lenguajes de transformación de documentos XML. En cuanto a los servicios web, se hizo una breve revisión de SOA, el protocolo SOAP, el lenguaje de descripción WSDL y el lenguaje de coreografías WS-CDL.

En cuanto a la evaluación socio-económica, se vieron algunos criterios de aprobación de políticas y proyectos públicos, y se vio que la metodología de evaluación social de proyectos se basa en medir el impacto de un proyecto en la función de bienestar social. Se vio que en la práctica se realiza el cálculo del valor presente neto de un proyecto, usando los precios sociales provistos por el MIDEPLAN. También se vio que, en casos complejos, se usan indicadores de rendimiento para comparar proyectos (IVAN, costo-beneficio y costo-impacto).

En cuanto a las métricas, se vio que las métricas en la ingeniería de software se encuentran en un estado muy inmaduro. Las métricas presentadas en la teoría tienen escasa credibilidad. Por eso, se hizo un breve formalismo para definir varios conceptos como el de medida, métrica, métrica directa y métrica indirecta. También se vieron criterios de validación de métricas, aunque tienen un carácter más o menos informal.

4 Modelo de análisis del gobierno electrónico

4.1 Introducción

En este capítulo se estudia el efecto que tiene el gobierno electrónico en lo que es la administración pública y el quehacer del gobierno, generando un marco global de análisis para los siguientes capítulos.

A modo de aclaración metodológica, este capítulo entrega el sustento teórico para analizar ordenadamente el gobierno electrónico, separando partes de distinta naturaleza. Ahora, el desarrollo de este capítulo está basado fuertemente en trabajos de expertos. Esto permite que la información presentada aquí actué, de alguna forma, como un pronóstico²⁸ a nivel cualitativo.

4.2 La administración pública actual

El gobierno es, básicamente, una entidad²⁹ administrativa muy grande, compuesta por muchas entidades más pequeñas de gran variedad. Sin embargo, es posible clasificar distintos aspectos de la administración pública para separar el análisis.

Según el nivel de decisiones, [Musso 2005] hace una diferenciación entre una entidad política y una entidad operacional:

- ENTIDAD DE CARÁCTER POLÍTICO: este tipo de entidad toma decisiones de carácter político o administrativo. Trabaja sobre aspectos globales; se centra en la planificación.
- ENTIDAD DE CARÁCTER OPERACIONAL: este tipo de entidad es de tipo técnico-operativa. Es ágil, toma decisiones a corto plazo e implementa los planes definidos por la entidad de carácter político, y le presta asesoría técnica.

Según la naturaleza estructural de la entidad, [Musso 2005] hace una diferenciación en la naturaleza de las tareas desarrolladas:

- ESTRUCTURA DE SERVICIO: está centrada en labores jurídico-administrativas. Este tipo de

28 Ésta es una versión barata del método *Delphi* de pronóstico cualitativo a largo plazo. En vez de reunir expertos, aquí se reúnen trabajos y opiniones.

29 Se usa la palabra *entidad* en su forma legal. Puede ser intercambiada por *institución* u *organismo* sin mayores problemas (nótese que este tipo de institución es una entidad).

entidad le *da la cara* a los ciudadanos. El SII (Servicio de Impuestos Internos) es un ejemplo representativo.

– TECNOESTRUCTURA: está vinculada a un campo específico y desarrolla una estructura particular en torno a éste. La Conama, el Ministerio de Salud, el Sernac, etc. son buenos ejemplos de tecnoestructuras.

A modo de observación, las tareas más *programables* están en las estructuras de servicio, a nivel operacional. Esto será de especial interés más adelante.

Otra separación, importante en cuanto a la orientación de la administración en sí misma (ya a nivel de *actitud* gubernamental) es:

– ADMINISTRACIÓN PÚBLICA TRADICIONAL: el estilo de administración que busca el buen funcionamiento de las entidades públicas.

– ADMINISTRACIÓN PÚBLICA PARA EL DESARROLLO: el estilo de administración que busca la generación de impacto en la ciudadanía, en pos de elevar su bienestar social.

También es importante notar que el modelo de trabajo de la administración pública es de tipo *burocrático*³⁰. Las entidades públicas están sujetas controles a legales y su proceder está normado. Además, el trabajo del sector público, que es de naturaleza administrativa, está centrado en el trabajo de la información. Todo esto incide en un intenso trabajo documental.

4.3 Tendencias en la modernización del estado

Las tendencias políticas suelen tener grandes efectos en las decisiones tomadas por los gobiernos. La modernización del estado también es una tendencia, aunque lleva gestándose hace muchos años; se puede decir que el proceso de modernización del estado partió con el mismo retorno de la democracia [Doña 2006]. La etapa actual de modernización es el *gobierno electrónico* en sí mismo, en donde la tecnología tiene un rol esencial. La etapa anterior, sin embargo, estaba centrada en el mejoramiento de la gestión de las personas y el rediseño de procesos, como se puede observar en [HACIENDA 1999].

Si bien la modernización es una tendencia, [Criado 2002] distingue dos tendencias divergentes en cuanto a enfoques:

30 Burocracia u *oficinismo*, en la misma definición entregada por Max Weber.

- LA NUEVA GESTIÓN PÚBLICA (NGP): Esta tendencia busca migrar las técnicas de gestión de la empresa privada a la administración pública. Busca la eficiencia. Es parte de la *nueva economía institucional*.
- LA GOBERNANZA PÚBLICA: Esta tendencia busca resaltar valores políticos como la participación ciudadana, la cuestión del medio ambiente, la ética política, etc. en contraposición a la Nueva Gestión Pública. A la Gobernanza Pública también se le conoce como *governabilidad pública*.

Casos particulares de estos enfoques aparecen dentro de las tendencias de lo que es la *administración electrónica* o *e-administración*. Estos enfoques son muy conocidos:

- E-GOVERNMENT O GOBIERNO ELECTRÓNICO: Este enfoque busca mejorar el desempeño del gobierno a través del uso de las tecnologías de la información.
- E-GOVERNANCE O GOBERNABILIDAD ELECTRÓNICA: Este enfoque busca potenciar los valores políticos a través de las tecnologías de la información.

Es evidente que el *gobierno electrónico* es a la *nueva gestión pública* como la *governabilidad electrónica* es a la *governanza pública*. Como se puede observar, los enfoques son complementarios: uno se enfoca en *cómo se hacen* las cosas mientras que el otro se enfoca en *qué hacer*. Sin embargo, [PRYME 2006] ha manifestado que no hay claridad en cómo avanzar en la *governabilidad electrónica*.

Respecto a los aspectos técnicos, el énfasis ha sido usar las tecnologías relacionadas con Internet para implementar el gobierno electrónico. Como se puede ver en [PRYME 2003] y [PRYME 2006], los esfuerzos se han centrado en:

1. La creación de páginas web para que el usuario³¹ acceda a los servicios provistos por el gobierno sin la necesidad de acudir físicamente a sus dependencias.
2. La automatización -en la medida de lo posible- del trabajo dentro de las entidades públicas, usando las tecnologías de la información.
3. La utilización del XML como formato oficial de la documentación electrónica.

31 El término *usuario* hace alusión a su naturaleza de consumidor (es quien *usa*). En la literatura aparece como *ciudadano*, *administrado*, *cliente*, *consumidor*, *beneficiario*, incluso -en voces anglosajonas- como *stakeholder* o *tax-payer* [Cornejo 2006]. La contraparte del usuario corresponde al *funcionario* o *servidor público*.

En una línea mucho más automatizada, el proyecto *Plataforma Integrada de Servicios Electrónicos del Estado* (PISEE) está centrado en la utilización de una arquitectura *SOA* (orientada al servicio) a través de *servicios web*³² que interactúan.

4.4 Metodología general de implementación

En general, la informatización del gobierno ha sido normativa. El Decreto Supremo 81 impuso el uso del XML como estándar para la documentación electrónica. En cuanto a metas, el Decreto Supremo 83 impuso plazos para la implementación de repositorios y del uso de documentos XML (administración, definición en XML *Schema*, etc.).

En cuanto a la atención a usuario, [Cornejo 2006] presenta las etapas del Programa de Mejoramiento de la Gestión para el gobierno electrónico:

1. La elaboración de un diagnóstico por parte de los organismos. Se presenta al PRYME.
2. La elaboración de un plan de mejoramiento que incluya el uso de las TIC en la provisión de sus *productos* estratégicos, procesos y canales tecnológicos de participación ciudadana y transparencia. Se presenta al PRYME.
3. La implementación del programa de trabajo de desarrollo del plan en 2.
4. La elaboración de un informe de resultados y recomendaciones que es presentado al PRYME.

Además de esto, muchos organismos han implementado otras acciones para mejorar la calidad de la atención, algunos con mucho éxito.

Respecto a las etapas del gobierno electrónico, no hay mayor consenso a nivel internacional; [PRYME 2006] indica que sólo hay dos etapas comunes: el inicio y el gobierno electrónico unificado (final). De todas maneras, el instructivo presidencial indica cuatro etapas:

1. PRESENCIA: Provisión de información de los servicios públicos al ciudadano.
2. INTERACCIÓN: Comunicación simple entre el servicio público y el ciudadano. Aparece el uso de esquemas de búsqueda.

³² Servicios web o *web services*. Son las unidades funcionales en una arquitectura orientada al servicio o SOA (*Service Oriented Architecture*).

3. TRANSACCIÓN: Las transacciones electrónicas se vuelven una alternativa a la atención presencial.

4. TRANSFORMACIÓN: Considera cambios en los servicios, de acuerdo a su misión electrónica, y la introducción de aplicaciones que administren la entrega de prestaciones al ciudadano.

A modo de observación, las etapas definidas por el instructivo presidencial están centradas en la atención al ciudadano. En otras palabras, es un gobierno electrónico *orientado al servicio*. Sin embargo, muchos de los objetivos del gobierno electrónico van más allá de eso, como se puede ver en [PRYME 2006]. El sentido parece ir por lo que es *generar bienestar inmediato* al ciudadano en vez de lograr una transformación absoluta hacia lo que es la administración electrónica. Esto es coherente con la claridad actual respecto a la dirección del gobierno electrónico. En cambio, no hay claridad respecto a cómo avanzar hacia la gobernabilidad electrónica.

4.5 Una clasificación para los flujos de información

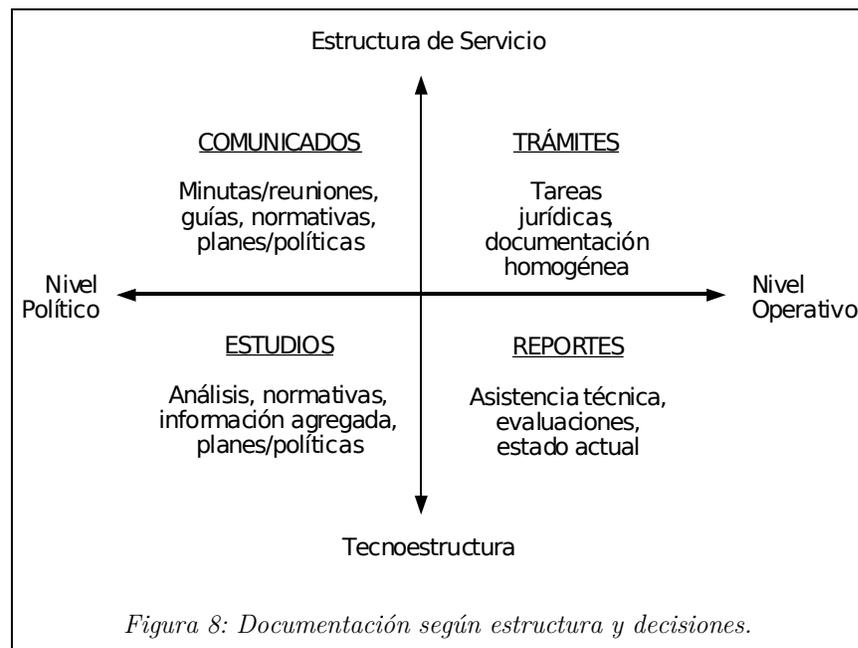
Considerando las clasificaciones mencionadas en [Musso 2005], se puede comenzar a dividir el tipo de flujo de información para un análisis posterior (que corresponde a la construcción de modelos matemáticos y la derivación de las métricas).

4.5.1 Clasificación del tipo de documentos según la naturaleza del trabajo

Es cierto que una entidad genera muchos tipos de documentación, y que tiene que ser muy especializada para generar un sólo tipo de ésta. Sin embargo, para evitar la complejidad del análisis, se hará *una propuesta*³³ *que generaliza la documentación según la naturaleza del trabajo*. Debido a que toda acción oficial debe quedar registrada o documentada, una clasificación de la naturaleza de la documentación también sirve para clasificar la naturaleza del trabajo.

La clasificación propuesta se puede ver en la figura 8. La división del tipo de documentación es muy general y pretende hacer énfasis en la naturaleza del trabajo realizado según la clasificación.

33 El modelo que se presentará responde a una de los objetivos de esta memoria.



EXPLICACIÓN DE LA FIGURA 8:

- *Nivel operativo*: la documentación generada es rutinaria.
 - *Estructura de servicio*: la información generada corresponde a **trámites**, que representan labores jurídico-administrativas. Reflejan la interacción con el usuario.
 - *Tecnoestructura*: se basa en informes rutinarios o **reportes** presentados a un nivel supervisor, sea de nivel directivo o ejecutivo.
- *Nivel político*: la documentación generada es más profunda y decisiva, se refiere a cosas de gran escala o grandes horizontes de tiempo.
 - *Estructura de servicio*: además de dar cuenta a otros niveles de tutela, se indican los cambios (extensiones, rediseños, políticas, etc.) al nivel operativo; en este sentido, la documentación estaría centrada en **comunicados**.
 - *Tecnoestructura*: toma decisiones de alto nivel, con información especializada, teniendo efecto en regulaciones y normativas; por su nivel especialista, su documentación estaría centrada en **estudios**.

Nota: Esta clasificación no es excluyente. Es normal la interacción de los cuadrantes.

4.5.2 Clasificación según la naturaleza programable del trabajo

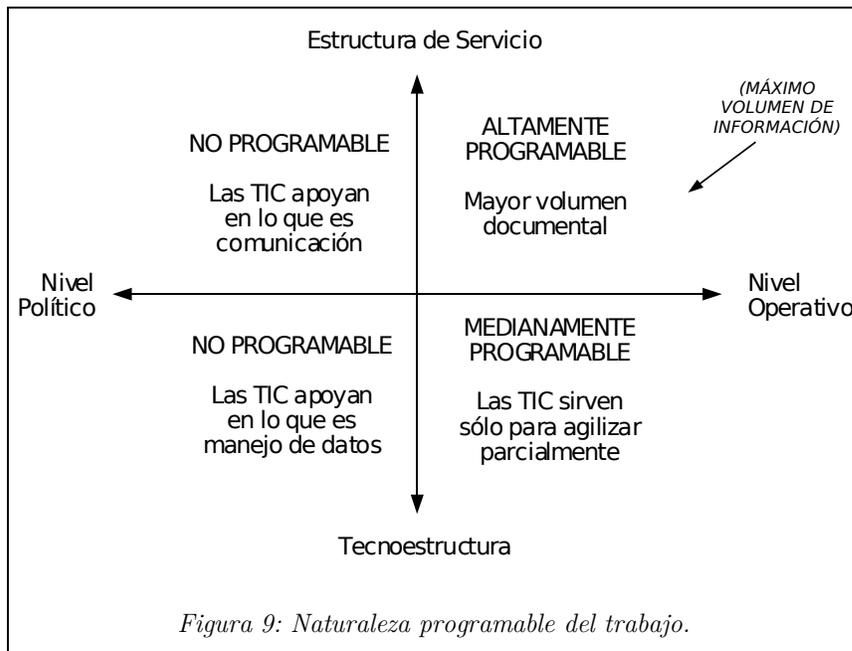
En lo que concierne a la naturaleza del trabajo, es posible observar lo siguiente:

- *Los trámites son la componente más automatizable del flujo documental.* Ello se debe a su naturaleza normada y rutinaria. Que sea normada significa que sus procedimientos están especificados por una ley. Que sea rutinaria implica que el procedimiento ocurre de forma repetitiva, con escasa novedad.

- *El nivel operacional de la tecnoestructura es medianamente programable* (documentación de tipo *reporte*). Ello se debe a su naturaleza rutinaria. La automatización va en el enfoque de la utilización de aplicaciones para aligerar la carga de trabajo³⁴.

- En el nivel político, en cambio, la automatización se reduce a la generación de información agregada para la supervisión y la toma de decisiones, y en la comunicación de políticas y cambios en el quehacer de la entidad. Es necesario notar que la toma de decisiones, pese a que pueda ser rutinaria (debido a la rutina de la reuniones, por nombrar un ejemplo), es un proceso reflexivo, lo cual dificulta su automatización.

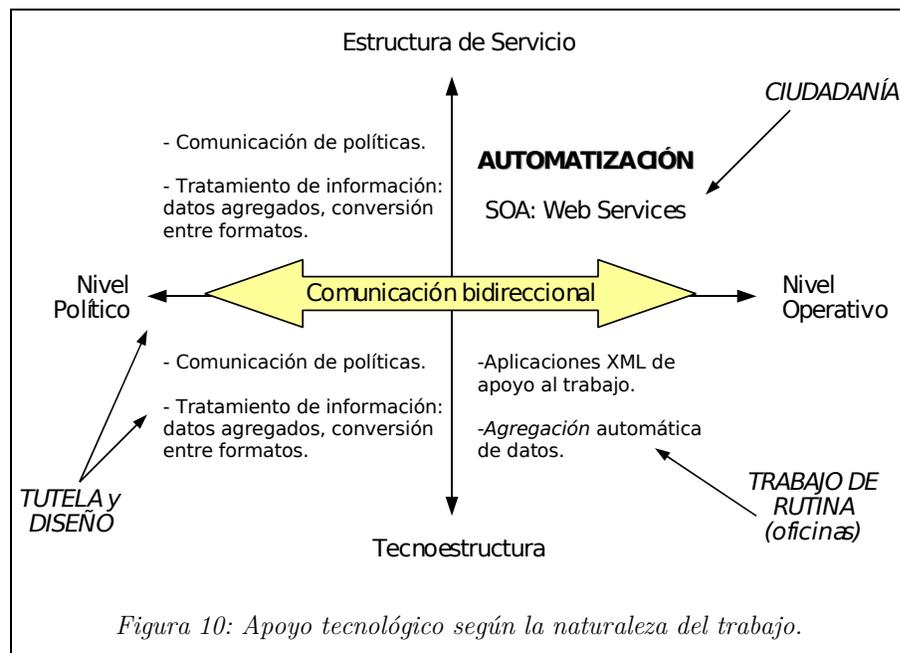
Todo lo anterior se sintetiza en la figura 9.



34 Estas aplicaciones están orientadas al lenguaje especialista de estas estructuras. Los sistemas expertos pueden ser un gran aporte en el cuadrante tecnoestructura-operativa.

4.5.3 Implementación del gobierno electrónico según la naturaleza del trabajo

Notando lo anterior, el nivel técnico del asunto se reduce a apoyar de la mejor manera la labor de las distintas entidades según su naturaleza. Idealmente, todo lo que se pueda programar será automatizado y cualquier aplicación de apoyo será desarrollada. Por lo visto en la sección de tendencias en la modernización del estado, distintas tecnologías serán aplicadas en cada caso. Por supuesto, el uso de XML es transversal a toda la clasificación de la figura 10.



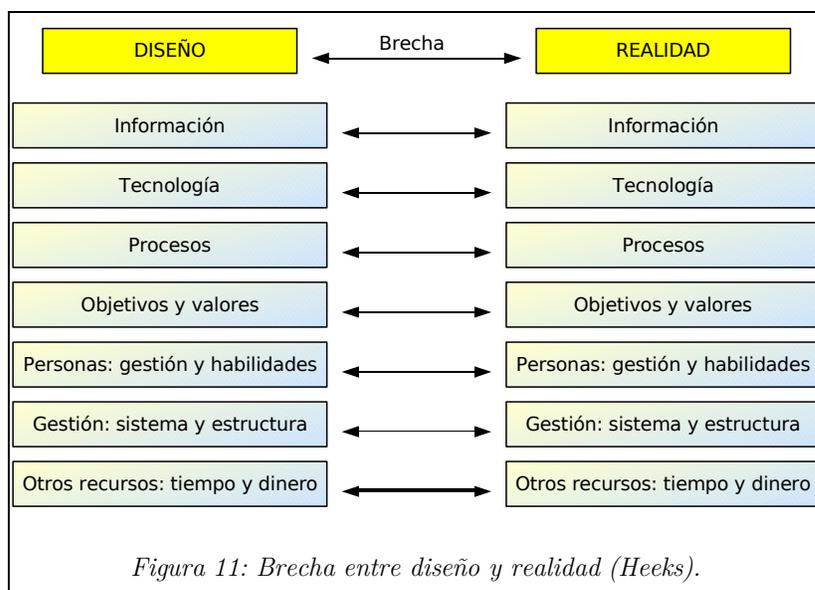
Los servicios, y la recuperación de la información generada por ellos, es la parte más automatizable del gobierno. Su presentación a las personas será preferentemente vía web. Sin embargo, tras esta capa de presentación funcionará una extensa red de servicios web que interactuarán entre sí.

En el resto de los casos, la parte humana sigue siendo prioritaria. En esos casos, el gobierno electrónico se manifestará como una infraestructura de soporte para sus labores diarias, con variados beneficios según el caso.

4.6 La situación real y la situación ideal

A nivel internacional, muchas iniciativas de gobierno electrónico han terminado en el fracaso. Esto se puede apreciar en [Heeks 2006], que argumenta que el fracaso de estas iniciativas

radica en un traspaso irreflexivo de las prácticas del sector privado al sector público. Las situaciones son diferentes, luego los diseños no coinciden con la realidad del sector público. Heeks presenta su modelo de auditoría *ITPOSMO* que es acrónimo de *Information, Technology, Process, Objectives and values, Staffing and skills, Management systems and structures* y *Other resources: time and money*. El modelo de la brecha se presenta en la figura 11 (adaptación personal).



En términos generales, ITPOSMO se puede usar como un apoyo al diseño. Sin embargo, se puede usar para formular una observación entre diferentes situaciones. Si bien no es necesario ver la diferencia entre el sector privado y el sector público -debido a las exigencias metodológicas del Programa de Mejoramiento de la Gestión-, sí es bueno ver la brecha entre la situación actual y la situación deseada.

Un aspecto de la situación real es que, pese a las exigencias normativas, las administraciones regionales se encuentran en un estado muy precario con respecto a la administración central. En términos de fases, según el instructivo presidencial, muchas están en la fase 1.

Otro aspecto de la situación real es la continua implementación de la versión electrónica de los servicios ofrecidos por los organismos públicos a la ciudadanía. Se hace difícil suponer que todos los servicios públicos están representados por servicios web. Pero los esfuerzos están orientados a seguir agregando servicios web a la red informática del estado.

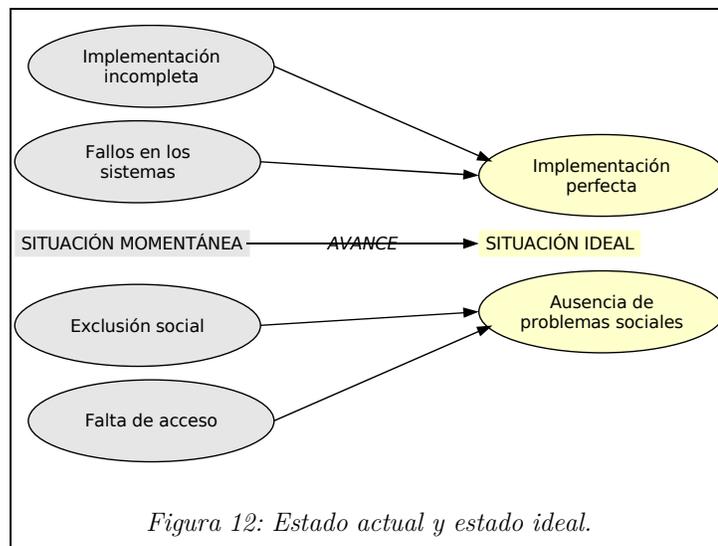
Por otro lado, existe una multitud de elementos sociales a considerar, como la alfabetización

digital [PRYME 2006]. La alfabetización digital es un requerimiento para el gobierno electrónico y es un tema que mezcla tópicos culturales y educacionales; [Raad 2006] ha hecho incapié en que la brecha digital -la analfabetización digital- puede ser un motivo de exclusión social, una fuente de desigualdad. [Raad 2006] plantea el problema específico de la “no contemporaneidad”, que consiste en que un individuo que ha quedado atrasado -excluido- en su alfabetización digital tiene que adaptarse a tecnologías que van cambiando, por lo que puede seguir atrasado al adaptarse a una tecnología obsoleta.

A manera de tratar de responder un desafío propuesto por [PRYME 2006]: ¿Cómo se puede evaluar el estado de avance del gobierno electrónico? Comparando la situación real con la situación ideal a la que se aspira; analizando una métrica para la brecha entre el diseño -la situación ideal- y la realidad. ¿Cómo es la situación ideal? Es posible describir la situación ideal así:

1. *Las tecnologías están perfectamente implementadas*: todo lo que es programable está automatizado por un servicio web o una aplicación, y está perfectamente integrado a los procesos de gobierno.
2. *No hay problemas sociales asociados al gobierno electrónico*: cualquier problema social representa una separación entre la realidad social actual y la realidad social ideal.

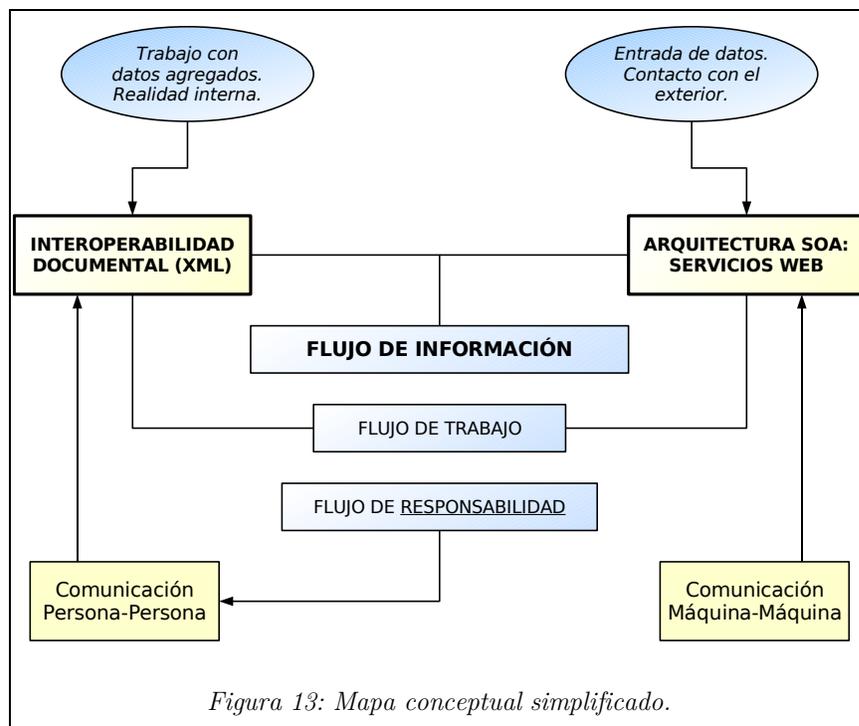
Cualquier contraste negativo significa un alejamiento de la situación ideal; reducir ese contraste implica un avance.



La figura 12 da una idea general de lo que podría ser un análisis de estado actual y estado ideal. Por supuesto, un análisis serio debería ser mucho más detallado. Al menos debería ser diferente entre un nivel *macro*, que debe ver problemas sociales a nivel general en la población, y un nivel *micro*, que debe contemplar sólo aquello que está al alcance de una entidad (como una municipalidad).

4.7 Un esquema general

De lo presentado y discutido en este capítulo, se ha visto que las tareas altamente programables son candidatas a convertirse en servicios web y que el resto de las tareas, menos programables, sólo pasan a ser apoyadas parcialmente por la tecnología. Esta diferencia abre dos posibilidades de modelamiento: la *interoperabilidad de los servicios web* y la *interoperabilidad documental* del trabajo de oficina (burocrático).



La figura 13 ilustra de forma sencilla los flujos de información representativos dentro de lo que es el gobierno electrónico. En lo que respecta a este trabajo, sólo se modelará el aspecto de los servicios web, debido a su naturaleza técnica e importancia en la percepción del ciudadano. La interoperabilidad documental, si bien tiene una gran componente técnica, también comprende la interacción del elemento humano, que es difícil de modelar de forma precisa.

EXPLICACIÓN DE LA FIGURA 13:

1. *El contacto con el mundo exterior está relacionado a los servicios web.* Esto se debe a que los ciudadanos (sean individuos o sociedades), serán usuarios de los sistemas de gobierno electrónico a través de los servicios web. Difícilmente habrá una interacción directa con los funcionarios públicos a través de documentos XML.

2. *Tanto los flujos de información como los flujos de trabajo están relacionados tanto con la interoperabilidad documental como con los servicios web.* En cuanto a los flujos de información, es claro que ambos aspectos están directamente relacionados a éstos. En cuanto a los flujos de trabajo³⁵, la relación a la interoperabilidad documental es directa: el manejo (generación, uso y transformación) de la documentación es parte de los flujos de trabajo. Los servicios web, por otro lado, permiten apoyar y gestionar flujos de trabajo.

3. *El flujo de responsabilidad sólo está asociado a la comunicación persona-persona.* Si un flujo de responsabilidad puede (y debiera) estar representado electrónicamente, no es un proceso automático. La delegación [Musso 2005] de responsabilidad o autoridad (de poder) es una tarea decidida por una persona, que recae sobre otra.

4. *La realidad interna está relacionada a la interoperabilidad documental.* Si bien los servicios web pueden apoyar el trabajo de una institución, incluso a nivel de oficina, no necesariamente representan una realidad interna. Los servicios web representan una funcionalidad accesible desde el exterior, o sea, una comunicación y una operación.

4.8 Resumen

En este capítulo se presentó un modelo de trabajo para el gobierno electrónico, con forma de cuadrantes. Este modelo está basado en una división funcional de los organismos públicos, según la naturaleza jerárquica del trabajo (nivel político y nivel operativo) y la orientación de éste (estructura de servicio y tecnoestructura). La combinación de las clasificaciones origina el modelo de cuatro cuadrantes.

Usando el modelo de cuadrantes, se observó cuan programables son los cuadrantes. El nivel político es el lado menos programable, debido a su naturaleza planificadora; el rol de las TIC es la supervisión y entrega de datos agregados, y el apoyo en las comunicaciones. El nivel operacional de la tecnoestructura es medianamente programable; el rol de las TIC es apoyar la labor de la tecnoestructura operativa a través de software especializado (gestión a través de flujos de

³⁵ Flujos de trabajo o *workflows*.

trabajos, utilización de sistemas expertos, apoyo a la generación de documentación típica en XML, etc.). El nivel operacional de la estructura de servicio, en cambio, es altamente programable debido a su naturaleza rutinaria y normada; el rol de las TIC es *automatizar* la rutina de los servicios, lo que se puede lograr bastante bien con los servicios web en una arquitectura SOA.

5 Métricas para las redes de servicios web

5.1 Introducción

Los servicios web tienen gran importancia en el gobierno electrónico:

- La tecnología de los servicios web permite automatizar el nivel operativo de la función de servicios de la administración pública (figuras 9 y 10, páginas 44 y 45 respectivamente).
- La interoperabilidad es una propiedad natural de los servicios web (especialmente si sigue un estándar como SOAP), que sea aprovechada bien con una arquitectura SOA (sección 3.3.2, página 17).
- El proyecto PISEE, *Plataforma Integrada de Servicios Electrónicos del Estado*, tiene por objetivo integrar los distintos servicios electrónicos del estado. Los servicios electrónicos contemplados son servicios web que siguen el estándar SOAP, en una arquitectura SOA [Gutiérrez 2005].

Debido a lo anterior, un correcto diseño, implementación y gestión de las redes de servicios web se torna muy importante. En este sentido, las técnicas de modelamiento y análisis de redes de servicios web presentadas en este capítulo pretenden ser un apoyo para los organismos públicos en el desarrollo de los servicios electrónicos.

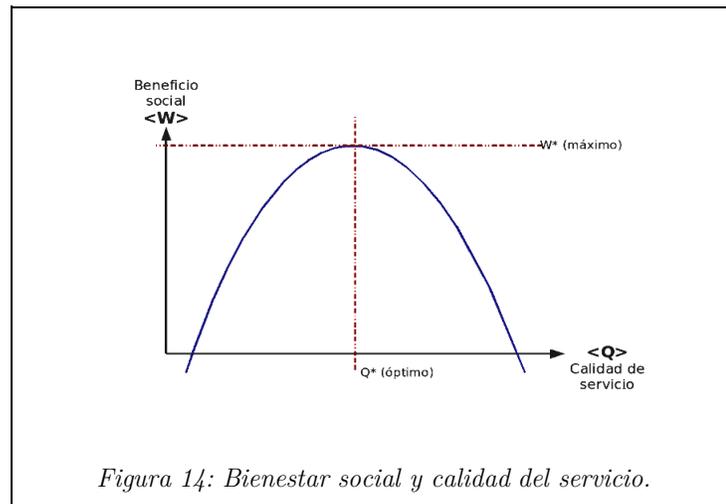
5.2 Consideraciones preliminares

5.2.1 Bienestar social y calidad del servicio

En teoría, el principal objetivo de la administración pública es maximizar el bienestar social³⁶. Un servicio público, como una manifestación particular de la administración pública, *hereda* este objetivo. Si se considera que la *calidad del servicio*³⁷ es la variable que determina el bienestar entregado por un servicio, entonces un servicio público buscará maximizar el bienestar social controlando la calidad del servicio. Este concepto se ilustra en la figura 14.

36 De este objetivo se pueden derivar muchos otros. Naturalmente, hay teorías alternativas sobre la administración pública.

37 La calidad del servicio es un término muy usado en la literatura industrial para caracterizar el valor entregado por un servicio.



Usando Q como la calidad del servicio, la función de bienestar social quedaría expresada conceptualmente como:

$$\text{BienestarSocial}(Q) = \text{BienestarEntregado}(Q) - \text{CostoProvision}(Q)$$

5.2.2 Función de provisión del servicio electrónico

En la teoría económica de la firma, una empresa busca maximizar su función de utilidad, que está expresada en términos de ingresos y costos. Usando principios de evaluación socio-económica, es posible expresar matemáticamente la *función de provisión* del servicio electrónico a través de información más desagregada que la calidad de servicio.

Primero, los “ingresos” de esta función de provisión se considerarán constantes. Un servicio es usado por su *funcionalidad*, no porque sea atractivo visualmente ni por otro motivo³⁸. Desde este punto de vista, si el servicio electrónico cumple con la funcionalidad deseada, entonces es capaz de satisfacer al cliente.

Segundo, el uso de un servicio electrónico puede significar un costo para el cliente. La calidad del servicio, en este caso, se reflejaría en un bajo costo. Reduciendo a términos más concretos, existe un costo asociado al acceso (interfaces poco comprensibles, uso de cibercafés, analfabetismo digital, etc.) y un costo de oportunidad asociado al tiempo de espera. Ambas dimensiones de costo se reflejan en el tiempo de uso de un servicio electrónico y en sensaciones psicológicas varias. De todas maneras, reducir todo a tiempo parece ser conveniente.

³⁸ Salvo, quizás, pruebas por parte del equipo de desarrollo.

Tercero, existen costos de provisión del servicio electrónico. Estos costos provienen del mantenimiento de los servicios y del consumo de los sistemas. Es importante notar que, debido a la naturaleza electrónica del servicio, el valor generado por un desarrollo se mantiene en el tiempo mientras que su costo ocurre en un instante de tiempo. Esto significa que el costo de provisión no tiene relación directa con la calidad del servicio entregado. Mas bien, la actividad sostenida (promedio) del mantenimiento de los servicios es la que explica el costo de provisión³⁹.

Notando que el “ingreso” es constante, se puede obviar de la función de provisión (dado que no influye en las decisiones). Entonces, la función de provisión se convierte en una función de costos. Naturalmente, el objetivo será minimizar los costos.

$$\begin{bmatrix} \text{Costo} \\ \text{Social} \end{bmatrix} = \begin{bmatrix} \text{Costo} \\ \text{Acceso} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Espera} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Provisión} \end{bmatrix}$$

El costo de acceso se controla a través de una mejora de las interfaces, del acceso a Internet y de la alfabetización digital. El costo de espera se controla mediante el *nivel de servicio* de los servicios web. El costo de provisión está en completo dominio del organismo.

Ahora, considerando el *costo social del tiempo*, el costo de acceso a Internet, el tiempo de *comprensión y uso de la interfaz* y la demanda, se puede reescribir la función, primero por el costo de uso:

$$\begin{bmatrix} \text{Costo} \\ \text{Ciudadano} \end{bmatrix} = \begin{bmatrix} \text{Acceso} \\ \text{Internet} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Tiempo} \end{bmatrix} \times \begin{bmatrix} \text{Comprensión} \\ \text{Interfaz} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Tiempo} \end{bmatrix} \times \begin{bmatrix} \text{Tiempo} \\ \text{Respuesta} \end{bmatrix}$$

Y la función de provisión del servicio:

$$\begin{bmatrix} \text{Costo} \\ \text{Social} \end{bmatrix} = \begin{bmatrix} \text{Costo} \\ \text{Ciudadano} \end{bmatrix} \times \begin{bmatrix} \text{Demanda} \\ \text{Servicio} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Provisión} \end{bmatrix}$$

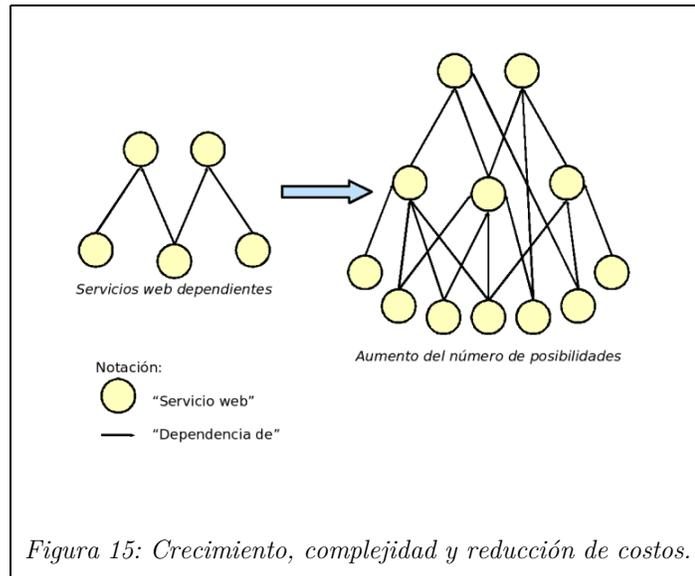
5.2.3 Efectos económicos de red

Los servicios interactúan en una red informática de servicios. Las consecuencias de este hecho son variadas y muy importantes.

AUMENTO DEL VALOR PARA TODOS LOS PARTICIPANTES. Cada vez que se agrega un servicio a la red, se facilita la implementación de nuevos servicios en base a éste y al resto. Las posibilidades

³⁹ Su labor no sólo debe consistir en vigilar. También debe haber énfasis en la mejora de los sistemas.

aumentan conforme al número de servicios ya implementados. A medida que la red crece, los costos de implementación se reducen y el crecimiento se acelera, permitiendo la implementación de servicios cada vez más complejos. Esto se ejemplifica en la figura 15.



ECONOMÍAS DE ESCALA. Por un lado, el costo marginal de agregar un nuevo servicio se reduce a medida que crece la red. Pero también ocurre que los costos marginales de provisión de un servicio, por usuario, son insignificantes. Es más cara la infraestructura que la atención de los usuarios.

BARRERAS DE SALIDA. Cuando un nuevo servicio web se agrega a la red, los servicios que se desarrollen en base a éste tendrán una **relación de dependencia**. Si la dependencia es exclusiva -no hay alternativas al servicio-, entonces todos los servicios web dependientes quedarán completamente inutilizados. Y los servicios web que dependan de los últimos también quedarán inutilizados. La sucesión es catastrófica. Este efecto genera una verdadera barrera de salida para los servicios.

(Ejemplo) En la figura 15, los dos servicios que encabezan la red son indispensables para el resto de los servicios. Si se quita uno de los dos, sólo tres servicios quedarán funcionales. Si se quitan ambos, la red pierde todo su valor.

Es bueno tener servicios web alternativos para reducir este riesgo de dependencia. Por supuesto, los servicios alternativos deben satisfacer los requerimientos de la dependencia para que sean alternativas viables. Así se evita una dependencia exclusiva.

5.3 Modelo de la red de servicios web

5.3.1 La red de dependencias

La red de servicios web se forma por cada servicio disponible para su utilización y las dependencias existentes entre sí. Naturalmente, no todos los servicios web estarán comunicados entre sí. De hecho, pueden haber servicios web completamente aislados del resto.

Definición. Se entenderá por **red de dependencias** a un grafo dirigido $G=(W,L)$, donde W es un conjunto de nodos o **servicios web** y $L \subset W \times W$ es un conjunto de arcos o **llamadas**, y un par $(a, b) \in L$ se lee “ a llama a b ” o “ a ejecuta a b ”.

La definición de este grafo dirigido (o *digrafo*) explica una instancia de una red. El conjunto de nodos W representa a todos los servicios web disponibles en una SOA. No importa que se identifique a cada servicio por palabras, números o signo. Si son palabras, W bien podría ser algo así: $W = \{A, B, C, Verificación, María, SAB, \dots\}$. Si son números, W bien podría ser algo así: $W = \{0, 1, 2, 3, \dots\}$. Si son signos: $W = \{\#, \sim, @, >, \neg, \dots\}$. Por supuesto, W tiene un número finito de elementos.

Por otro lado, L es un conjunto de arcos (conectan dos nodos en un grafo) que representa las llamadas en tiempo de ejecución. Por ejemplo, si un servicio web w_1 llama al servicio w_2 durante su ejecución, entonces $(w_1, w_2) \in L$. Como esto no necesariamente implica que $(w_2, w_1) \in L$, entonces el grafo es dirigido.

Una llamada en tiempo de ejecución representa una dependencia. Por ejemplo, si $(a, b) \in L$, cada vez que se ejecute a , el servicio web b también deberá ejecutarse. De hecho, b es parte de la ejecución de a (en este ejemplo). Si b no responde, entonces a tampoco puede responder. Esto implica una dependencia.

Sin embargo, si $\exists a, b, c \in W$ tales que $(a, b), (b, c) \in L$, se tiene que si a se ejecuta, entonces a llama a b y luego b llama a c . Si c no responde, entonces b no lo hace. Y si b no lo hace, a tampoco puede hacerlo. En este sentido, a depende de c . Pero $(a, c) \notin L$, o sea, a no llama a c (directamente).

Esto permite comenzar a definir formalmente una relación de **dependencia**, que indica que la ejecución de un servicio web implica la ejecución de otro. Así, la relación de dependencia aparece: si el servicio a implica la ejecución del servicio b , entonces a necesita de b para su funcionamiento. Si b no termina correctamente (arroja un error, por ejemplo), entonces a tampoco

puede hacerlo.

Definición. Un camino dirigido es una n -tupla (w_1, w_2, \dots, w_n) , $\forall i \in \{1, \dots, n\}, w_i \in W$, donde n es el largo del camino, que cumple $\forall i < n, (w_i, w_{i+1}) \in L$.

Propiedades. Se tienen las siguientes propiedades inmediatas de la definición anterior:

1. Toda llamada $(a, b) \in L$ es un camino dirigido, por consecuencia directa de la definición de este último.
2. Todo subcamino $(w_i, w_{i+1}, \dots, w_j)$ con $1 \leq i < j \leq n$ de un camino dirigido (w_1, \dots, w_n) es también un camino dirigido. Prueba: $\forall k = i, j-1, (w_k, w_{k+1}) \in L$ debido a que (w_1, \dots, w_n) es un camino dirigido, luego $(w_i, w_{i+1}, \dots, w_j)$ también es un camino dirigido.

Un camino dirigido representa una sucesión de arcos. En este caso particular, un camino dirigido representa una sucesión de llamados que provocan dependencia entre sí. Por ejemplo, si a llama a b y b llama a c , entonces (a, b, c) es un camino.

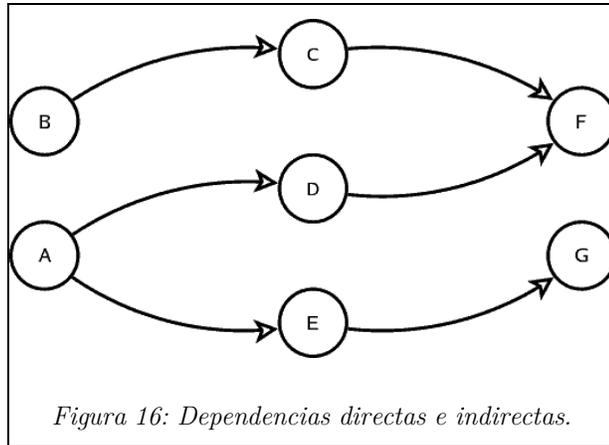


Figura 16: Dependencias directas e indirectas.

En la figura 16, A depende de G porque si A se ejecuta, G también lo debe hacer. Pero, por otro lado, B no depende de G para ejecutarse. De hecho, su ejecución no tiene relación con la ejecución de G ; no hay ningún camino o secuencia de llamados que relacione B con G . Otro alcance: G no depende de A para su ejecución. No existe ninguna secuencia de llamadas desde G que termine en A ; no existe ningún camino dirigido entre G y A .

Observación. Toda dependencia está representada por un camino dirigido.

A partir de la observación anterior es posible definir formalmente los conceptos de **dependencia**, **dependencia directa**, **dependencia indirecta** y **dependencia mixta** desde la definición de un camino.

Definición. Se define la **relación de dependencia** de la red de dependencias $G=(W,L)$ como el conjunto de pares $D\subseteq W\times W$ tal que $\forall (a,b)\in D$ existe un camino (a,\dots,b) . Por comodidad de notación, se define la función (relación) $Depende: W\times W\rightarrow\{V,F\}$ tal que $Depende(a,b)\Leftrightarrow(a,b)\in D$.

Definición. Una **dependencia directa** es una dependencia $(a,b)\in L$ y se lee “a depende directamente de b” o -in extenso- “el servicio web a depende directamente del servicio web b”. Esto es equivalente a “a llama a b”. Por notación, se denotará $Directa: W\times W\rightarrow\{V,F\}$, con $Directa(a,b)\Leftrightarrow(a,b)\in L$.

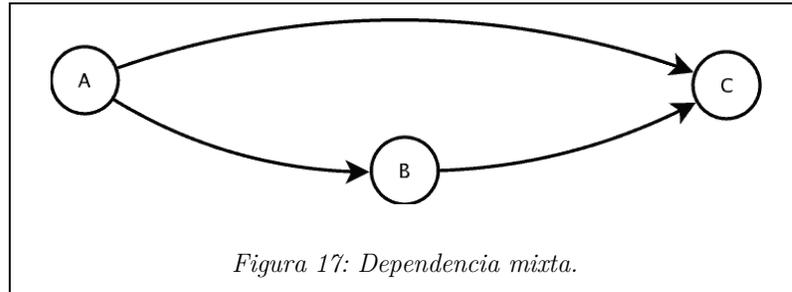
Definición. Una **dependencia indirecta** es una dependencia $(a,c)\in D$ que cumple con la siguiente propiedad: $\exists b\in W,(a,b)\in D\wedge(b,c)\in D$ (transitividad). Por notación, se denotará $Indirecta: W\times W\rightarrow\{V,F\}$, con $Indirecta(a,c)\Leftrightarrow\exists b\in W,Depende(a,b)\wedge Depende(b,c)$

(Ejemplo) La figura 16 es un ejemplo de dependencias. Es un grafo dirigido descrito por un conjunto de nodos (servicios web) $W=\{A,B,C,D,E,F,G\}$ y un conjunto de arcos (llamadas) $L=\{(A,D),(A,E),(B,C),(C,F),(D,F),(E,G)\}$. El servicio web A depende directamente de D y E, e indirectamente de F y G (basta seguir las flechas). El servicio web B depende directamente de C e indirectamente de F. El servicio web C sólo depende de F. El servicio web D sólo depende de F. El servicio web E sólo depende de G. Estas tres últimas dependencias son todas directas.

Propiedades. Las relaciones de dependencias cumplen las siguientes propiedades:

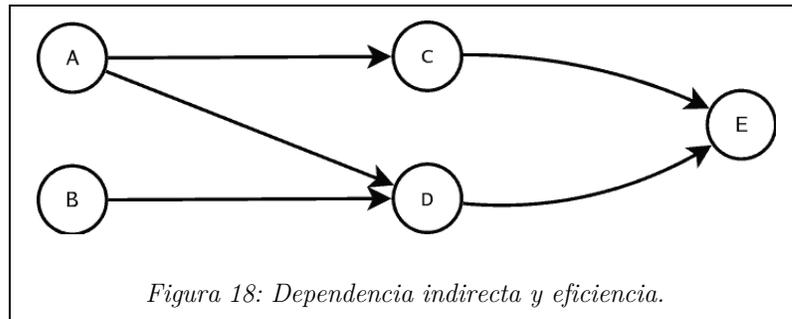
1. $Directa(a,b)\Rightarrow Depende(a,b)$, toda dependencia directa es una dependencia. Prueba: Una dependencia directa representa una llamada, pero toda llamada es un camino dirigido y todo camino dirigido es una dependencia. Se concluye que toda dependencia directa es una dependencia.
2. $Indirecta(a,b)\Rightarrow Depende(a,b)$, toda dependencia indirecta es una dependencia. Prueba:
 1. Por definición, $Indirecta(a,b)\Rightarrow\exists c\in W,Depende(a,c)\wedge Depende(c,b)$.
 2. Por definición de $Depende$, $\exists(w_1^1,w_2^1,\dots,w_n^1),(w_1^2,w_2^2,\dots,w_m^2)$, caminos dirigidos, tales que $w_1^1=a, w_n^1=c, w_1^2=c, w_m^2=b$.

3. Entonces se define $(w_1^3, \dots, w_{n+m-1}^3)$, con $w_k^3 = w_k^1$ si $k \leq n$ y $w_k^3 = w_{k-n+1}^2$ si $k \geq n$.
 4. Si $k < n$, entonces $(w_k^3, w_{k+1}^3) = (w_k^1, w_{k+1}^1) \Rightarrow (w_k^3, w_{k+1}^3) \in L$ pues (w_1^1, \dots, w_n^1) es un camino dirigido. Si $k \geq n$, entonces $(w_k^3, w_{k+1}^3) = (w_{k-n+1}^2, w_{k-n}^2)$, y se vuelve a cumplir que $(w_k^3, w_{k+1}^3) \in L$. Luego, $(w_1^3, \dots, w_{n+m-1}^3)$ es un camino dirigido.
 5. Como $(w_1^3, \dots, w_{n+m-1}^3)$ y $w_1^3 = a, w_{n+m-1}^3 = b$, se tiene que existe un camino dirigido entre a y b ; luego $(a, b) \in D$; $\text{Depende}(a, b)$.
 6. Se concluye que $\text{Indirecta}(a, b) \Rightarrow \text{Depende}(a, b)$.
3. $\text{Depende}(a, b) \Rightarrow \text{Directa}(a, b) \vee \text{Indirecta}(a, b)$, toda dependencia es una dependencia directa o una dependencia indirecta. Prueba:
1. $\text{Depende}(a, b) \Rightarrow \exists (w_1, \dots, w_n)$, un camino dirigido de largo n (por definición).
 2. Si $n=2$, $(w_1, w_2) \in L \Rightarrow \text{Directa}(a, b)$.
 3. Si $n > 2$, es posible construir dos subcaminos: (w_1, w_2) y (w_2, \dots, w_n) . Como ambos son subcaminos de un camino dirigido, también son caminos dirigidos. Luego, se cumple $\text{Depende}(w_1, w_2)$ y $\text{Depende}(w_2, w_n)$. Ahora, $w_1 = a$ y $w_n = b$. Entonces, se tiene que $\text{Depende}(a, b) \Rightarrow \text{Depende}(a, w_2) \wedge \text{Depende}(w_2, b)$. Por ende, se cumple que $\exists c \in W, \text{Depende}(a, c) \wedge \text{Depende}(c, b) \Rightarrow \text{Indirecta}(a, b)$, por definición.
 4. Como para $n=2$ se cumple $\text{Depende}(a, b) \Rightarrow \text{Directa}(a, b)$ y para $n > 2$ se cumple $\text{Depende}(a, b) \Rightarrow \text{Indirecta}(a, b)$, entonces siempre se cumple la condición más general $\text{Depende}(a, b) \Rightarrow \text{Directa}(a, b) \vee \text{Indirecta}(a, b)$. Luego se concluye que toda dependencia es directa o indirecta.
4. $\text{Depende}(a, b) \Leftrightarrow \text{Directa}(a, b) \vee \text{Indirecta}(a, b)$, toda dependencia es una dependencia directa o una dependencia indirecta, y toda dependencia directa o indirecta es una dependencia también. La prueba es trivial: sale de las propiedades anteriores.



La figura 17 muestra un caso en que A depende de C . Como se puede observar, A depende directamente de C (el arco superior). Pero también depende indirectamente a través del camino dirigido (A, B, C) . Ésta es una **dependencia mixta**.

Definición. Una **dependencia mixta** es una dependencia que es directa e indirecta. Se denotará, por comodidad, $Mixta: W \times W \rightarrow \{V, F\}$ la función (relación) definida como $Mixta(a, b) \Leftrightarrow Directa(a, b) \wedge Indirecta(a, b)$.



Una dependencia mixta refleja una potencial ineficiencia en la red: que un servicio web ejecute más de una vez otro servicio web, de manera redundante. Esto también puede ocurrir en dependencias puramente indirectas, pero el servicio web dependiente está menos involucrado en la ineficiencia. En la figura 18 se puede apreciar que A depende indirectamente de E a través de los caminos dirigidos (A, C, E) y (A, D, E) . Esto podría simbolizar una ineficiencia: quizás E podría ser llamado sólo una vez. Sin embargo, B depende de E sin presentar esta ineficiencia potencial. Se puede ver que la potencial ineficiencia es más bien relativa al nodo dependiente. En la dependencia mixta, en cambio, el servicio web dependiente es un actor directo en la ineficiencia. Por lo tanto, la potencial ineficiencia es indisoluble de la dependencia mixta.

5.3.2 Requerimientos de calidad de la red

¿Puede un servicio web llamarse a sí mismo? Sí, en principio. Pero no es conveniente. De

hecho, es absolutamente innecesario; es una llamada recursiva. Se puede implementar al interior del servicio web, sin generar tráfico. ¿Puede haber un ciclo dirigido en este grafo? Sí, en principio. Pero es totalmente inconveniente. De hecho, relaciona excesivamente el funcionamiento de varios servicios web, como si su implementación hubiera sido hecha simultáneamente. Eso va justo en contra del “espíritu” de la arquitectura SOA: separar completamente las funcionalidades, sin importar el funcionamiento de cada servicio en particular (cada servicio es una *caja negra*). Además, un ciclo dirigido puede derivar en un *deadlock*, en que los servicios tratan de terminar esperando al otro y viceversa, y la espera se anida, evitando que los servicios terminen (este es un escenario muy negativo y revelaría una gravísima falta por parte de los equipos de desarrollo). Entonces, por eficiencia y buenas prácticas de desarrollo, se supondrá que no se puede.

Definición. Un **ciclo dirigido** es un camino (w_1, \dots, w_n) tal que $w_1 = w_n$, o sea, hace a un servicio web dependiente de sí mismo: $\text{Depende}(w_1, w_1)$.

Observación. (Buenas prácticas de implementación) Sea D la relación de dependencias de la red de dependencias $G=(W,L)$. Entonces, $\forall (a,b) \in D \Rightarrow (b,a) \notin D$, o de forma equivalente, $\text{Depende}(a,b) \Rightarrow \neg \text{Depende}(b,a)$.

La propiedad anterior evita la existencia de ciclos dirigidos de cualquier tipo (el grafo será acíclico), incluso los servicios web que se llaman a sí mismos (el grafo no será un pseudografo).

Propiedades. Toda relación de dependencia que cumple $\forall (a,b) \in D \Rightarrow (b,a) \notin D$, también cumple las siguientes propiedades:

1. No hay llamadas reflejas en L : $\forall a \in W : (a,a) \notin L$. (Aplicación directa del supuesto)
2. No hay ciclos en dependencias directas: $\forall (a,b) \in L \Rightarrow (b,a) \notin L$. Tampoco se cumple un caso más general: $\forall (a,b) \in D \Rightarrow (b,a) \notin L$. (Aplicaciones directas del supuesto)
3. Si $w \in W$ tiene sólo una dependencia, o sea, $\forall u,v \in W : (w,u), (w,v) \in D \Rightarrow u=v$, entonces la dependencia es directa. Prueba: Sea $(w,u) \in D$ la única relación de dependencia de w . Si no es directa, $(w,u) \notin L$. Usando la definición de relación de dependencia, deberá existir $c \in W$ tal que $\{(w,c), (c,u)\} \in D$. Por lo tanto, w tendría al menos tres dependencias (todas diferentes), lo que es una contradicción. Luego, si un servicio web tiene sólo una dependencia, ésta será directa.

El cálculo de ciclos dirigidos puede ser una labor muy costosa. ¿Cómo asegurarse de construir una arquitectura SOA sin ciclos dirigidos? Siguiendo alguna buena práctica. En el caso de los servicios web, es bueno recordar que cada servicio es una unidad funcional independiente

del resto. Se propone, entonces, que todos los desarrollos de servicios web se hagan de manera independiente. O sea, que cada equipo de desarrollo sólo se enfoque en un servicio web a la vez, considerando los demás servicios como una caja negra. No importa que los desarrollos se hagan en paralelo por distintos equipos o uno detrás de otro. Lo que importa es que, siguiendo esta estrategia sencilla, no se desarrollen dependencias entre servicios web durante el desarrollo, de forma simultánea.

Observación. (Estrategia de desarrollo) *Un desarrollo de un servicio web que no toma en cuenta proyectos en marcha, sólo considera los servicios web terminados para basar su trabajo. Entonces, este servicio web sólo podrá ser dependiente de servicios web terminados.*

Propiedad. (Estrategia de desarrollo) *La observación anterior implica que la estrategia de desarrollo no genera ciclos dirigidos.*

Demostración. Supóngase que los servicios web se liberan uno a la vez (la liberación de varios servicios web se puede modelar como la liberación de un servicio web a la vez con tiempo cero entre liberaciones, gracias a la independencia de desarrollos). Sea $G_n = (W_n, L_n)$ la red de dependencias con n -liberaciones (hay n servicios web) y sea D_n la relación de dependencia asociada a G_n . En este escenario, la siguiente inducción ocurre:

1. CASO BASE: $G_1 = (w_1, \emptyset)$. Como no hay dependencias en el grafo, tampoco hay ciclos.
2. HIPÓTESIS INDUCTIVA: Sea G_n tal que D_n no contiene ciclos dirigidos (cumple con la observación sobre las buenas prácticas de desarrollo).
3. PASO INDUCTIVO: Sea w_{n+1} el último servicio web liberado y sean G_n y D_n el estado anterior de la red de dependencias. Por hipótesis inductiva, D_n no contiene ciclos dirigidos. Entonces:

1. Si w_{n+1} no tiene dependencias: $W_{n+1} = W_n \cup \{w_{n+1}\}$, $L_{n+1} = L_n$. Así, $D_{n+1} = D_n$, por lo que $G_{n+1} = (W_{n+1}, L_{n+1})$ no tiene ciclos dirigidos.
2. Si w_{n+1} tiene dependencias: $\forall (v_1, \dots, v_m)$ camino dirigido existente en G_n , $v_k \neq w_{n+1}$, $\forall k=1, m$. debido a que w_{n+1} no estaba en W_n . Ahora, supóngase que en G_{n+1} hay un ciclo dirigido. Si w_{n+1} es parte del ciclo, entonces existe un camino $(v_1, \dots, v_k, w_{n+1}, v_{k+2}, \dots, v_r)$, con $v_1 = v_r$. Pero esto implica que la llamada $(v_k, w_{n+1}) \in L_{n+1}$, o sea, había un servicio web dependiente a uno no terminado, una

contradicción. Entonces w_{n+1} no es parte de ningún ciclo. Pero esto significa que el ciclo (v_1, \dots, v_r) estaba presente en G_n debido a que $(v_k, v_{k+1}) \in L_n, k < r$, lo que es una contradicción respecto a la hipótesis inductiva. Luego no se generan ciclos si w_{n+1} tiene dependencias.

3. Se concluye que si D_n no tiene ciclos dirigidos, entonces D_{n+1} tampoco los tendrá.

Esto permite concluir que, con prácticas sencillas de desarrollo, se pueden evitar ciclos en la red. Como los ciclos (dirigidos) son muy nocivos en este tipo de red, **se supondrá que no hay ciclos en la red**. El supuesto parece bien admisible pues, si la red llegara a tener ciclos, la calidad del servicio se volvería extremadamente mala; basta con imaginar cómo se sobrecargarían los sistemas si hubieran ciclos y cómo se dilatarían los tiempos de respuesta de los servicios web.

5.4 Métricas y calidad de servicio

La calidad del servicio se puede entender como una combinación entre el nivel de servicio -la probabilidad de satisfacer un requerimiento en un cierto período de tiempo- y ciertos atributos del servicio entregado. Esto entrega dos planos complementarios para lograr una cierta calidad de servicio: uno cuantitativo y uno cualitativo (en la medida que se quiera tratar de esa manera).

En lo que sigue, se propondrán distintas métricas para enfrentar los requerimientos de calidad de servicio, como apoyo para la toma de decisiones.

5.4.1 Controlando el nivel de servicio

Estimar el nivel de servicio de un servicio web, dado un *tiempo exigido* de respuesta, es una labor bastante sencilla. Sin embargo, controlar el nivel de servicio puede tornarse una tarea bastante complicada debido a la gran cantidad de factores que lo determinan. ¿Qué factores pueden afectar el nivel de servicio? A continuación se enumera una breve lista:

- Una implementación *subóptima* de uno o más servicios web.
 - Un algoritmo ineficiente puede restar rendimiento a un servicio.
 - Malas prácticas de programación podrían incidir en un uso excesivo de tiempo y otros recursos de un sistema.
- La sobrecarga del servidor, a causa de una gran demanda de sus servicios web,

podría reducir el rendimiento de cada servicio.

- Hay servicios que son demasiado demandados.
- Dependencias ineficientes; un servicio web podría estar siendo solicitado de manera redundante.
- Hay servicios que son cruciales; si su rendimiento decae, gran parte de la red también lo hará.
- Vaivenes de la conexión.
- Un servidor ha dejado de funcionar, y así sus servicios.
 - El servidor se ha dañado debido a una falla externa, por ejemplo, debido a la subida de la tensión eléctrica, a un corte de suministro eléctrico, a un recalentamiento, etc.
 - Sabotaje (ataque por parte de *sombreros negros*⁴⁰).
- Un servicio web podría tornarse lento a causa de algún servicio del cual dependa.
- Un servidor tiene una potencia insuficiente; está obsoleto.
- Un servicio web se ha construido con una tecnología obsoleta⁴¹.

Si un servicio web no está siendo lo suficientemente rápido o confiable (su nivel de servicio es inferior al deseado), entonces se hace necesario determinar las medidas a tomar. En lo que concierne a estas líneas, las métricas a presentar están destinadas a determinar sobre qué servicios web destinar los esfuerzos para mejorar el nivel de servicio. Se da menor énfasis en lo que respecta a las acciones a tomar, pero tampoco se dejan de lado.

5.4.2 Midiendo el nivel de servicio

A modo de recomendación, parece razonable definir los servicios web más importantes en términos de nivel de servicio. A modo de ejemplo, los servicios web más demandados por la

40 Un sombrero negro es un hácker malicioso (crácker) que destruye sistemas informáticos.

41 Una tecnología obsoleta no debería afectar el nivel de servicio. Sin embargo, los costos de obsolescencia están relacionados con un costo de cambio: el mantenimiento y la actualización.

ciudadanía parecen más importantes que el resto. Luego, el nivel de servicio de estos servicios web parece más relevante que el de otros servicios. Si estos servicios prioritarios tienen un nivel de servicio insuficiente, un estudio de las causas de la ineficiencia debiera partir de éstos. Así se comenzaría a revisar la red, en la búsqueda de posibles aspectos a mejorar.

¿Cómo determinar si un servicio web cumple con un determinado nivel de servicio? A saber, se necesita tener un tiempo exigido y una probabilidad mínima de cumplimiento. Si la probabilidad de cumplir es menor a la mínima, no se está cumpliendo con el nivel de servicio deseado. Naturalmente, exigir demasiado puede llevar a grandes costos innecesarios mientras que exigir muy poco puede llevar a tener una calidad de servicio deficiente.

Para determinar la probabilidad de cumplir con un determinado tiempo exigido, se presentan un par de métricas: una orientada a una revisión de un período de tiempo y otra orientada a una recopilación automática.

Definición. Se define el **tiempo de respuesta** de un servicio web como el tiempo tomado entre que se realiza la petición al servicio hasta que el servicio entrega una respuesta satisfactoria.

Definición[Número]. Se define la función **Nro**, o “número”, $Nro: A \rightarrow \mathbf{N}_0$ como la cardinalidad de A , siendo A un conjunto finito.

Definición[Nivel de Servicio]. Se define la métrica **NServicio**, $NServicio: W \rightarrow [0,1]$, de un servicio web como el cociente entre el número de veces que el tiempo de respuesta (T^w) es menor al tiempo exigido (T_0) y todas las peticiones al servicio. Se denotará la métrica nivel de servicio como un mapeo de un servicio web al intervalo cerrado $[0,1]$:

$$NServicio(w) = \frac{Nro(T^w < T_0; \text{en período } t)}{Nro(\text{Peticiones}^w; \text{en período } t)} \approx \wp(T^w < T_0)$$

La métrica **NServicio** mide la probabilidad de que un servicio web cumpla un tiempo exigido T_0 . La correctitud de la métrica es axiomática; es acorde a la definición de probabilidad.

Definición[Nivel de Servicio Suavizado]. Se define la métrica **NServS**, $NServS: W \rightarrow [0,1]$, como la función recursiva:

$$NServS(w) \leftarrow (1 - \lambda) NServS'(w) + \lambda [\text{si } T^w < T_0]$$

donde $NServS'$ es el valor anterior de la métrica y λ es un parámetro a ajustar entre 0 y 1.

*NServS se actualiza cada vez que el servicio web w se ha ejecutado. El valor inicial de *NServS* debe estar en ley ($0 \leq \text{NServS} \leq 1$).*

Propiedad. *El valor esperado de *NServS* es $\wp(T^w < T_0)$.*

Prueba. Esperanza: $E(\text{NServS}) \leftarrow (1-\lambda)E(\text{NServS}') + \lambda \wp(T^w < T_0)$. Sea $S_n = E(\text{NServS})$, calculado n -veces, y $\wp = \wp(T^w < T_0)$. Luego: $S_n = (1-\lambda)S_{n-1} + \lambda \wp$. Ahora, sea $D_n = |S_n - \wp|$, otra sucesión. Expandiendo, tenemos:

$$\begin{aligned} D_n &= |S_n - \wp| = |(1-\lambda)S_{n-1} + \lambda \wp - \wp| = |(1-\lambda)S_{n-1} - (1-\lambda)\wp| \\ &\Rightarrow D_n = (1-\lambda)|S_{n-1} - \wp| = (1-\lambda)D_{n-1} \end{aligned}$$

Es trivial ver que $D_n = (1-\lambda)^n D_0$ -por inducción-. Como $(1-\lambda) \in (0,1)$, entonces la sucesión converge a cero. Por lo tanto, S_n converge a \wp , o sea, $E(\text{NServS}) \rightarrow \wp(T^w < T_0)$.

Propiedad. *La métrica *NServS* es un estimador con las siguientes propiedades:*

1. *El error esperado de la n -ésima iteración de *NServS* es inferior a $(1-\lambda)^n$. Prueba: Usando la sucesión $D_n = |S_n - \wp| = (1-\lambda)^n D_0$, definida en la prueba anterior, y considerando que $0 \leq D_0 \leq 1$, se tiene que $D_n = (1-\lambda)^n D_0 \leq (1-\lambda)^n$.*
2. *En el límite, la varianza de *NServS* es $\text{Var}(\text{NServS}) = \frac{\lambda \wp}{2-\lambda} > 0$. Prueba: Tomando varianza a la definición de *NServS*, y considerando límite (la varianza ha convergido), se tiene: $\text{Var}(\text{NServS}) = (1-\lambda)^2 \text{Var}(\text{NServS}) + \lambda^2 \wp \Rightarrow (2\lambda - \lambda^2) \text{Var}(\text{NServS}) = \lambda^2 \wp$
 $\Rightarrow \text{Var}(\text{NServS}) = \frac{\lambda \wp}{2-\lambda}$.*
3. *En el límite, la varianza de *NServS* es creciente según λ . Prueba: Retomando el resultado anterior, $\text{Var}(\text{NServS}) = \frac{\lambda \wp}{2-\lambda}$, se ve claramente que la función es creciente según λ : el numerador es creciente y el denominador es decreciente.*

La métrica *NServS* sirve para una estimación en una actualización automática. La constante λ debe ser elegida con cuidado: un valor muy cercano a cero tiene una velocidad de convergencia baja y un valor cercano a uno hace inestable a la métrica. Como esta métrica está

pensada para un cálculo automatizado, la velocidad de convergencia no parece ser un problema pues irían a ocurrir muchas iteraciones en poco tiempo (cientos o miles por mes⁴²).

¿Qué valores asumir para el parámetro λ y para el valor inicial de la métrica NServS? Suponiendo que se espera un buen nivel de servicio, NServS debería valer 1 al principio. λ debería elegirse pensando en lograr una desviación estándar⁴³ pequeña. Si el nivel de servicio es alto ($\rho \approx 1$), se puede hacer la siguiente aproximación: $Var(NServS) \approx \frac{\lambda}{2-\lambda}$. En términos de desviación estándar: $\sigma(NServS) \approx \sqrt{\frac{\lambda}{2-\lambda}}$. Exigiendo una desviación σ , se puede despejar λ : $(2-\lambda)\sigma^2 = \lambda \Rightarrow \lambda = \frac{2\sigma^2}{1+\sigma^2}$. Por ejemplo, si se exige una desviación estándar de 0,01, el valor de λ que satisface esto es $2 \cdot 10^{-4}$.

Como observación, las métricas NServicio y NServS buscan estimar el mismo valor: $\rho(T^w < T_0)$. Sin embargo, la métrica NServicio debiera ser usada cuando se hace un análisis ocasional del nivel de servicio. En cambio, la métrica NServS debiera usarse cuando se desea seguir de cerca⁴⁴ el nivel de servicio.

5.4.3 Métricas estructurales

Las métricas estructurales se basan en la estructura de la red de dependencias para medir distintos aspectos de los servicios web (o de la red misma). Estas métricas son algo estáticas debido a su naturaleza: se basan en la forma de la red. Por ende, no necesitan ser actualizadas frecuentemente (a menos que la estructura de la red cambie muy seguido). Tampoco consideran llamadas condicionales (no siempre ocurren) o múltiples (cuando un servicio web llama varias veces a otro) entre servicios web. Sólo consideran la forma de la red.

Para empezar, se definirá un par métricas directas: “Entrantes” y “Salientes”.

Definición[Entrantes]. Se define la métrica **Entrantes** de un servicio web w , *Entrantes*: $W \rightarrow \mathbf{N}$, como el número de servicios web que acceden directamente a w , o sea:

42 [PRYME 2006] muestra la demanda de los servicios en línea. Cientos o miles de accesos mensuales parece una suposición razonable (incluso convencional).

43 La desviación estándar es la raíz de la varianza, y es especialmente útil porque conserva las unidades de la variable aleatoria original.

44 Si se desea usar un *cuadro de mando*, NServS es preferible a NServicio.

$$\begin{aligned} Entrantes(w) &= Nro(v \in W : (v, w) \in L) \Leftrightarrow \\ Entrantes(w) &= Nro(v \in W : Directa(v, w)) \end{aligned}$$

Definición[Salientes]. Se define la métrica **Salientes** de un servicio web w , $Salientes : W \rightarrow \mathbf{N}$, como el número de servicios web llamados por w , o sea:

$$\begin{aligned} Salientes(w) &= Nro(v \in W : (w, v) \in L) \Leftrightarrow \\ Salientes(w) &= Nro(v \in W : Directa(w, v)) \end{aligned}$$

(Ejemplo) La figura 19 muestra un servicio web W que tiene tres entrantes (los servicios que llaman a W) y dos salientes (los servicios que son llamados por W).

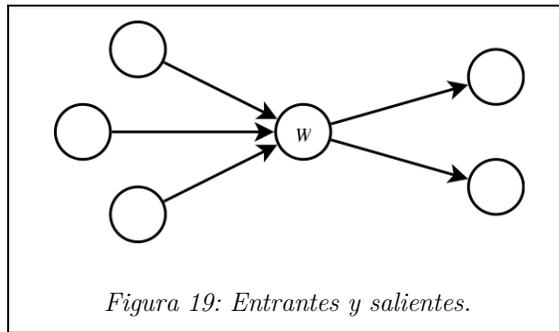


Figura 19: Entrantes y salientes.

¿Cuánta carga genera un servicio web al ejecutarse? En otras palabras, ¿cuántas llamadas implica la ejecución de un servicio web? Suponiendo que cada dependencia representa una llamada⁴⁵, cada vez que un servicio web se ejecuta, sus dependencias directas son todas ejecutadas también, y al ejecutarse, las llamadas de cada una también son ejecutadas, recursivamente. Por ejemplo, si a llama a b y b genera 10 llamadas, entonces a genera la llamada a b y las 10 llamadas que produce b .

Definición[Carga Estructural Generada]. Se define la métrica **CargaSGen** de un servicio web w , $CargaSGen : W \rightarrow \mathbf{N}$, como el número de llamadas implicadas por la ejecución de w (la carga generada). Sumando una llamada por dependencia directa y la carga generada por éstas, se define recursivamente:

$$CargaSGen(w) = \sum_{v \in W : Directa(w, v)} (1 + CargaSGen(v))$$

Observaciones:

45 En esta sección sólo se considera la estructura -la forma- de la red, no su ejecución real. Por lo tanto, no se puede considerar la probabilidad de usar una dependencia: realizar una llamada. Se asume que ocurre.

- La fórmula tiene en cuenta que llamada se ejecuta una vez. La carga producida por una llamada es producto de ejecutar el servicio web llamado y de la carga que genera ese servicio al resto de la red (recursividad).
- Si w no depende directamente de otro nodo, entonces no ocurre la suma. Luego, la métrica $CargaSGen$ será simplemente 0.

Si un servicio web genera mucha carga en una red, está teniendo muchas dependencias. Este es un potencial riesgo de una arquitectura SOA: la dependencia excesiva. Ahora, ¿cuánta dependencia hay en la red? Observando que la red tiene una cantidad máxima de dependencias directas posibles ($n(n-1), n \stackrel{\text{def}}{=} |W|$), es posible construir una razón que refleje el grado de dependencia en la red.

Definición[Grado de Conectividad]. Se define la métrica **GrConex** de la red $G(W,L)$, $GrConex : G \rightarrow [0,1]$, como:

$$GrConex(G) = \frac{Nro(L)}{Nro(W) \times (Nro(W) - 1)}$$

Observación. Un grafo completo -el que tiene a todos sus nodos conectados- posee $n(n-1)$ vértices, siendo n el número de nodos. Luego, $GrConex$ es la razón entre el número actual de vértices (llamadas) del grafo y el número máximo que admite.

La métrica $GrConex$, *grado de conectividad* o *densidad*, indica cuan conectada está la red. A mayor número de conexiones, más cercano a 1 será su valor. Una conexión total vale 1. Un valor alto, cercano a 1, indicaría una red demasiado dependiente. Por el contrario, un valor muy cercano a cero indicaría un escaso uso de la red.

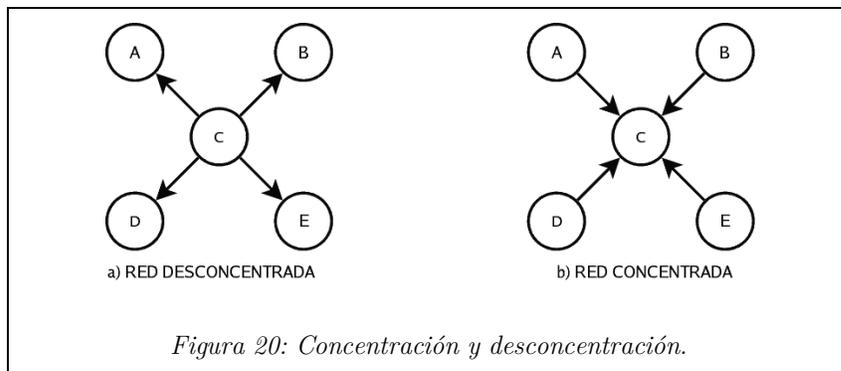


Figura 20: Concentración y desconcentración.

$GrConex$ es insuficiente para indicar aspectos de la red relacionados con la orientación de

las dependencias; de hecho, si se define $L' = \{(a, b) \in W \times W : (b, a) \in L\}$, se ve que la red de dependencias $G = (W, L')$ tiene el mismo grado de conectividad que $G = (W, L)$, pero todos los arcos están invertidos. La figura 20 muestra que la dirección de los arcos es muy importante. En la red desconcentrada (a), no parece haber un nodo más importante que otro. Sin embargo, si se cae cualquier servicio web, C se caería también. Además, si C se ejecuta, toda la red se ejecuta también. En la red concentrada (b), hay un nodo que es crítico para el funcionamiento de la red: C. Si C se cae, toda la red queda inutilizada. Pero si se cae A, B, D o E, el resto de la red sigue funcional.

La noción informal de red concentración de la red, es que si algunos nodos determinan el funcionamiento de gran parte de la red, la red está concentrada en esos nodos (depende de esos nodos). En cambio, si los nodos dependen de muchos otros, la red está desconcentrada. Aquí aparece un *trade off*: una red concentrada es muy sensible a la caída de ciertos servicios web mientras que una red desconcentrada tiene servicios que generan mucha carga al resto de la red. Ahora, ambos problemas desaparecen si la red está desconectada, o sea, tiene un GrConex nulo (o pequeño). Y si GrConex es alto (cercano a uno), estos problemas se agravan.

Definición[Grado de Concentración]. Se define la métrica **GrConcen** de una red $G(W, L)$, $GrConcen : G \rightarrow [0, 1)$, como:

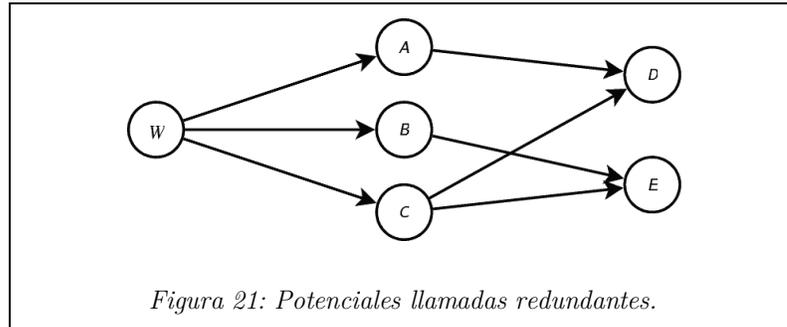
$$GrConcen(G) = \sqrt[6]{\frac{\sum_{w \in W} Entrantes^2(w)}{2Nro^3(W) - 3Nro^2(W) + Nro(W)}}$$

Definición[Grado de Desconcentración]. Se define la métrica **GrDesconcen** de la red $G(W, L)$, $GrDesconcen : G \rightarrow [0, 1)$, como:

$$GrDesconcen(G) = \sqrt[6]{\frac{\sum_{w \in W} Salientes^2(w)}{2Nro^3(W) - 3Nro^2(W) + Nro(W)}}$$

Observación. El origen de las fórmulas es algo complicado, por lo que se explica en el apéndice A.

(Ejemplo) Retomando el ejemplo de la figura 20, para la red desconcentrada (a) se tiene: $GrConex(a) = 4/20 = 0,2$, $GrConcen(a) = 4/20 = 0,2$ y $GrDesconcen(a) = 2/20 = 0,1$. Para la red (b), $GrConex(b) = 4/20 = 0,2$, $GrConcen(b) = 2/20 = 0,1$ y $GrDesconcen(b) = 4/20 = 0,2$. Se puede ver que, si bien GrConex es indiferente al orden de las dependencias, GrConcen y GrDesconcen crecen con la concentración o desconcentración de la red, respectivamente.



Otro aspecto relativo a la eficiencia es el número de servicios web que se deben ejecutar. Esto es diferente al número de llamadas que implica la ejecución de un servicio (medido por CargaGen). La razón: la redundancia. La figura 21 muestra un servicio web W que implica la ejecución de cinco servicios web (A, B, C, D, E), pero ocurren siete llamadas. Los servicios D y E son ejecutados dos veces. Esto puede ser una potencial redundancia.

Definición[Carga Subestimada]. Se define la métrica **CargaSub** de un servicio web w , $CargaSub: W \rightarrow \mathbf{N}$, como el número de servicios web de los que w depende. $CargaSub$ se calcula como:

$$CargaSub(w) = Nro(v \in W : Dependee(v, w))$$

Definición[Recarga]. Se define la métrica **ReCarga** de un servicio web w , $ReCarga: W \rightarrow \mathbf{N}$, como:

$$ReCarga(w) = CargaSGen(w) - CargaSub(w)$$

La métrica **ReCarga** indica la cantidad potencial de ejecuciones redundantes que son consecuencia de la ejecución de un servicio web. Esta métrica debe ser usada con cuidado pues no indica realmente si las potenciales redundancias son redundancias reales. En este sentido, **ReCarga** sólo sirve para iniciar una inspección.

5.4.4 Métricas de funcionamiento

En esta sección se presentan métricas que miden aspectos del funcionamiento de los servicios web. Este tipo de métricas debiera ser actualizada regularmente. La manera más sencilla de hacer esto es a través del *minado* de los logs.

Un primer atributo medible es la calidad de servicio, que ya ha sido abarcado debido a su prioridad. Todo el análisis parte de vigilar el nivel de servicio. Y se presentaron dos métricas:

NServicio y NServS calculadas por:

$$NServicio(w) = \frac{Nro(T^w < T_0; \text{en período } t)}{Nro(Peticiones^w; \text{en período } t)} \approx \wp(T^w < T_0)$$

$$NServS(w) \leftarrow (1 - \lambda) NServS'(w) + \lambda [\text{si } T^w < T_0]$$

Un segundo atributo medible es la aleatoriedad de las dependencias. Una dependencia puede no ser usada como una llamada a un servicio web. Y, si es usada, puede que signifique la realización de varias llamadas. En este sentido, dos métricas debieran ser consideradas: la probabilidad de realizar una llamada y el número esperado de llamadas que realiza un servicio realiza ante otro, por ejecución. Una observación inmediata: si un servicio web llama a otro servicio web varias veces durante una ejecución, puede ser síntoma de llamadas redundantes. Un análisis del motivo de las múltiples llamadas podría revelar una oportunidad de mejora de un servicio, debido a la menor carga generada durante su ejecución y al menor tiempo de respuesta.

Definición[Probabilidad de Llamada]. Se define la métrica **Pr** de una llamada, $Pr : L \rightarrow (0, 1]$, como la probabilidad que la llamada $(a, b) \in L$ sea realizada durante una ejecución del servicio web a . Por complemento:

$$Pr((a, b)) = \frac{N - Nro(\text{Veces que } a \text{ no llama a } b, \text{ en las últimas } N \text{ ejecuciones})}{N}$$

Observación. $1 - Pr(l)$ es la probabilidad que una ejecución de un servicio web no use la llamada l . La fórmula usada para calcular Pr es, por ende, $Pr = 1 - (1 - Pr)$. La validez de $1 - Pr(l)$ es axiomática; obedece la definición frecuentista de probabilidad.

Definición[Número de Llamadas]. Se define la métrica **Nl** de una llamada, $Nl : L \rightarrow \mathbb{R}^+$, como el número esperado de veces que un servicio a realiza una llamada $(a, b) \in L$, por ejecución.

$$Nl((a, b)) = \frac{Nro(\text{llamadas de } a \text{ a } b, \text{ últimas } N \text{ ejecuciones de } a)}{N}$$

Observación. La fórmula para calcular Nl es simplemente una media muestral (un promedio), un tradicional estimador de la esperanza.

Por comodidad, se usará equivalentemente $Pr(a, b) = Pr((a, b))$ y $Nl(a, b) = Nl((a, b))$. Esta notación alternativa no entrega problemas.

Volviendo al tema de la potencial redundancia asociada al número de llamadas a un servicio web, es posible notar que una llamada l que ocurre a lo más una vez por ejecución, tiene la propiedad: $Pr(l) = Nl(l), l \in L$. De lo contrario, $Pr(l) < Nl(l)$. De hecho, Nl se puede relacionar a Pr de la siguiente forma: $Nl(a, b) = E(\text{Veces que } a \text{ llama a } b | a \text{ necesita de } b) \times Pr(a, b)$. Si el servicio web a necesita de la operación provista por b , realizar la llamada más de una vez puede significar redundancia. Esto, naturalmente, dependerá de cada caso, pero definir una métrica para detectar este tipo de casos puede ser de utilidad.

Definición[Número Relativo de Llamadas]. Se define la métrica **Nlr** de una llamada l , $Nlr: L \rightarrow \mathbb{R}^+ \cup \{0\}$, como el “sobreuso” (suponiendo que un servicio web debiera llamar a otro servicio web a lo más una vez) de llamadas que realiza un servicio web sobre l . La métrica **Nlr** se calcula como:

$$Nlr(l) = \frac{Nl(l)}{Pr(l)} - 1$$

Observación. La razón Nl/Pr estima la cantidad esperada de ejecuciones de l que un servicio hace, dado que va a usar la dependencia l . La filosofía es que, si va a hacer una llamada, hacerla más de una vez puede significar redundancia. Para completar, se resta uno, dejando sólo la potencial redundancia esperada del uso de la llamada l .

Si **Nlr** es igual a cero, entonces la llamada no ocurre de manera redundante. Si es mayor que cero, *puede* significar que hay redundancia (depende de la naturaleza de las llamadas). Debido al uso de valores estadísticos, esta métrica es capaz de detectar sólo aquellas redundancias que se manifiestan en la práctica. Esto puede ser útil al momento de buscar qué mejorar en una red; si **Nlr** es positivo, entonces hay una posibilidad de reducir el número llamadas que realiza un servicio, mejorando su tiempo de respuesta.

El tiempo de respuesta de un servicio web está fuertemente relacionado al nivel de servicio y a la aleatoriedad de las dependencias. El tiempo de respuesta se había definido como el tiempo transcurrido entre que el servicio web recibe una petición y hasta que el servicio emite una respuesta satisfactoria (incluso si esta respuesta es nula, o sea, el servicio sólo termina su ejecución de forma correcta). Pero, ¿qué información se puede obtener del tiempo de respuesta? Por una parte, permite conocer cuánto se demora realmente el servicio web sin considerar la carga generada (descontando el tiempo que significa usar las dependencias). Por otra parte, si además se dispone de su varianza, permite estimar el nivel de servicio para distintos niveles de exigencia.

Definición[Tiempo Esperado de Respuesta]. Se define la métrica **TRespuesta** de un servicio web w , $TRespuesta: W \rightarrow \mathbb{R}^+$, como el tiempo promedio de respuesta del servicio web.

$T_{\text{Respuesta}}$ se calcula como:

$$T_{\text{Respuesta}}(w) = \frac{1}{N} \sum_{k \in \{\text{Últimas } N \text{ ejecuciones}\}} T_k^w$$

$T_k^w \stackrel{\text{def}}{=} \text{Tiempo de respuesta del servicio web } w \text{ en su } k\text{-ésima ejecución}$

Definición[Varianza del Tiempo de Respuesta]. Se define la métrica $V_{\text{Respuesta}}$ de un servicio web w , $V_{\text{Respuesta}}: W \rightarrow \mathbb{R}^+$, como la varianza del tiempo de respuesta del servicio web. $V_{\text{Respuesta}}$ se calcula como:

$$V_{\text{Respuesta}}(w) = \sum_{k \in \{\text{Últimas } N \text{ ejecuciones}\}} \frac{\left(T_k^w - T_{\text{Respuesta}}(w)\right)^2}{N-1}$$

Observación. La fórmula para calcular $V_{\text{Respuesta}}$ corresponde a una varianza muestral.

Definición[Tiempo Propio de Respuesta]. Se define la métrica $P_{\text{Respuesta}}$ de un servicio web w , $P_{\text{Respuesta}}: W \rightarrow \mathbb{R}^+$, como el tiempo “propio” de respuesta, aquel no explicado por las llamadas realizadas por el servicio web. $P_{\text{Respuesta}}$ se calcula como:

$$P_{\text{Respuesta}}(w) = T_{\text{Respuesta}}(w) - \sum_{(w,v) \in L} Nl(w,v) \times T_{\text{Respuesta}}(v)$$

Observación. Asumiendo que un servicio web debe esperar la respuesta de los servicios que solicita, descontar el tiempo de espera revela el propio tiempo de proceso del servicio.

El uso conjunto de $T_{\text{Respuesta}}$ y $V_{\text{Respuesta}}$ permite estimar niveles de servicios para distintas exigencias de tiempo. Usando una función de densidad normal⁴⁶ de media $T_{\text{Respuesta}}$ y varianza $V_{\text{Respuesta}}$, se puede realizar la estimación. $P_{\text{Respuesta}}$, en cambio, sirve para descubrir oportunidades de mejoras de implementación a nivel de algoritmos o malas prácticas de programación. Si $P_{\text{Respuesta}}$ es pequeño, esa oportunidad es reducida (¿cuánto es posible mejorar algo que demora un milisegundo?). Entonces, un $P_{\text{Respuesta}}$ grande presenta una potencial oportunidad de mejora de la implementación del servicio web.

Otro atributo medible -muy importante por cierto- es la demanda de un servicio web. La demanda representa cuántas solicitudes (llamadas) se realizan a un servicio en un determinado período de tiempo. En este sentido, la demanda es una frecuencia.

46 Esta estimación se vuelve especialmente confiable si se usa un N grande para el muestreo.

Definición[Demanda Total]. Se define la métrica **Demanda** de un servicio web w , $Demanda: W \rightarrow \mathbb{N}$, como la frecuencia de accesos al servicio web (sea o no proveniente de la red) en un intervalo de tiempo I .

$$Demanda_I(w) = Nro(\text{Accesos a } w \text{ en el intervalo } I)$$

Se podría considerar que usar “último mes” como valor del intervalo I en la definición pudiera conducir a alguna ambigüedad. Pero no hay ambigüedad de fecha en los sistemas informáticos⁴⁷. Si el mes es Enero y se observa la métrica en lo que va del mes, la métrica sólo debiera contar los accesos de Enero. En este sentido, la métrica Demanda es acumulativa en el número de accesos dentro de un intervalo de tiempo. En lo que sigue, se omitirá la identificación de los intervalos de tiempo en las métricas de demanda. La elección del intervalo de tiempo queda a cargo del evaluador.

La demanda de un servicio web es información relevante. Y es un buen criterio para definir servicios web cuyo nivel de servicio es relevante. Pero esta información se puede desagregar: es posible determinar si el servicio web tiene alta demanda *interna a la red* o alta demanda *externa a la red*. La demanda interna es aquella originada por otros servicios web. Un servicio con alta demanda interna es crucial en la red. La demanda externa es aquella que no proviene de otros servicios web (de la red $G(W,L)$); el nivel de servicio de estos servicios web explica directamente la calidad de la atención electrónica (o virtual) al ciudadano. Por eso, conviene medir la demanda interna y la demanda externa.

Definición[Demanda Interna]. Se define la métrica **DdaInterna** del servicio web w , $DdaInterna: W \rightarrow \mathbb{R}$, como la demanda explicada por las llamadas realizadas desde el interior de la red. $DdaInterna$ se puede estimar como:

$$DdaInterna(w) \approx \sum_{v \in W: \text{Directa}(v,w)} Demanda(v) \times Nl(v,w)$$

Definición[Demanda Externa]. Se define la métrica **DdaExterna** del servicio web w , $DdaExterna: W \rightarrow \mathbb{R}$, como la demanda que no es explicada por las llamadas realizadas al interior de la red. $DdaExterna$ se calcula como:

$$DdaExterna(w) = Demanda(w) - DdaInterna(w)$$

La carga de un servicio web (y de su servidor) proviene principalmente de su demanda.

⁴⁷ A veces hay, como en los cambios de hora. Pero el impacto de este tipo de imprecisiones es menor en el momento de hacer estadísticas.

Pero un servicio web también puede generar demanda a otros servicios. Esta demanda representa la carga de trabajo, sobre la red, que provoca la ejecución de ese servicio web. Ya se había definido, en las métricas estructurales, una carga generada que no consideraba información de ejecución de los servicios web. Ahora, se definirá la carga real generada sobre la red, considerando aspectos de la ejecución.

Definición[Carga Real Generada]. Se define la métrica **CargaRGen** del servicio web w , $CargaRGen: W \rightarrow \mathbb{R}^+$, como la demanda total generada sobre la red, por una ejecución del servicio. $CargaRGen$ se calcula como:

$$CargaRGen(w) = \sum_{(w,v) \in L} Nl(w,v) \times (1 + CargaRGen(v))$$

$CargaRGen$ mide el número real de llamadas generadas por la simple ejecución de un servicio web. Si se multiplica por la demanda de ese servicio, el resultado será toda la demanda ocasionada por el servicio web, ya sea a través de dependencias directas o indirectas.

Otra métrica muy importante, es la *disponibilidad* del servicio. La disponibilidad se define normalmente como la proporción del tiempo que un bien o servicio se puede usar. El resto del tiempo, el servicio está dado de baja o el bien se ha estropeado. Considerando el caso de los servicios web, la demanda diaria de un servicio puede variar regularmente (ser más alta los fines de semana, durante el mediodía, etc.) por lo que la disponibilidad deja de tener sentido. Lo que realmente interesa es la probabilidad de que un servicio web responda correctamente una petición.

Definición[Probabilidad de Servicio Disponible]. Se define la métrica **PrDisponible** de un servicio web w , $PrDisponible: W \rightarrow (0,1]$, como la probabilidad de que un servicio web se responda correctamente. $PrDisponible$ se calcula como:

$$PrDisponible(w) = \frac{Nro(w \text{ ha respondido correctamente, últimas } N \text{ ejecuciones})}{N}$$

$PrDisponible$ debe ser muestreada por un período largo de tiempo (N debe ser muy grande). Un servicio web no queda inutilizado con frecuencia... Los sistemas informáticos actuales son bastante resistentes a las caídas. Por eso, $PrDisponible$ debe ser muestreada durante un período largo de tiempo, para incorporar varias caídas dentro de las observaciones. De todas maneras, se debe cumplir que: $PrDisponible(w) > NServicio(w)$ (pues toda caída reduce el nivel de servicio). La métrica $PrDisponible$ se debe mantener alta, al igual que $NServicio$, igual o casi igual a 1. Valores como 0,95 son muy bajos para $PrDisponible$; considerando los objetivos del gobierno electrónico, el propósito del servicio web es estar disponible en todo momento.

5.5 Modelamiento de una red

5.5.1 Modelamiento desde una situación real

CONSTRUCCIÓN DE LA ESTRUCTURA DE LA RED

Como se aprecia en la figura 22, para construir el grafo hay que revisar cada servicio web presente en la red. Por cada servicio web se debe agregar un nodo al grafo, al conjunto W de servicios web.

Al analizar un servicio web, hay que registrar todas las llamadas que realiza. Cada una de esas llamadas debe ser agregada al conjunto L de llamadas. Así se tendrán los arcos del grafo.

Una vez construidos W y L , es posible definir la red de dependencias $G=(W,L)$. Por supuesto, esto no es suficiente para realizar todos los análisis sobre la red de servicios web. El grafo G sólo representa las dependencias.

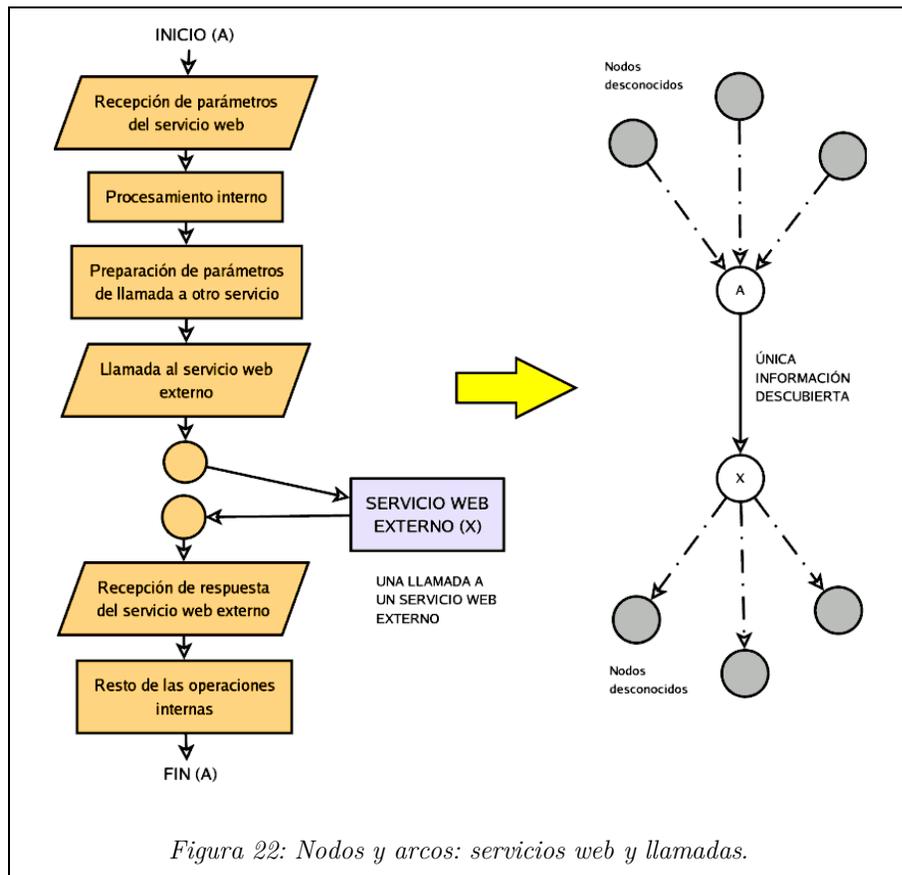


Figura 22: Nodos y arcos: servicios web y llamadas.

¿Qué otra información recopilar? Dependiendo de las métricas que se deseen evaluar, se debe recopilar toda la información necesaria para éstas. También depende de la cuan seguido se desee conocer la información de la red; en ese caso, será necesario implementar algún sistema de monitoreo. De todas maneras, mucha de esta información se puede obtener de los *logs*⁴⁸ de cada servidor.

ALIMENTACIÓN DE LAS MÉTRICAS

Las métricas estructurales no necesitan ser actualizadas regularmente. Sólo cuando la *estructura* de la red es alterada (un cambio en W o en L). Cada vez que un servicio web adquiere la madurez suficiente para ser incluido en la red, la red cambia su estructura. Cada vez que un servicio web es rediseñado en cuanto a sus dependencias, la red cambia su estructura. Cada vez que un servicio web es removido, la red cambia su estructura. Y así sucesivamente.

Diferente es el caso de las métricas de funcionamiento. Éstas requieren información estadística, por lo que deben ser actualizadas con cierta frecuencia. Se recomienda alimentar las métricas con información obtenida de los logs del servidor.

EL CASO DE UNA RED LIMITADA

No siempre se debería modelar toda la red para hacer un análisis parcial de la situación. La decisión de modelar sólo una porción de la red se puede deber, entre otros, a los siguientes motivos:

- *Dominio*. Si se desea evaluar el estado de la red que está a cargo de un determinado organismo.
- *Alcance de información*. La información que no está disponible, se debiera omitir. No se pueden tomar decisiones sobre elementos desconocidos.

Una red limitada debería modelarse de la misma manera que las otras redes, sólo que evitando los servicios web fuera del dominio a modelar. ¿Por qué hacer esto? Porque el uso de información fuera del dominio de decisión no debiera afectar las decisiones tomadas sobre la red propia.

48 *Log*: registro de la actividad del servidor. Incluye información como peticiones html, respuestas y tiempos.

5.5.2 Modelamiento desde una situación hipotética

DISEÑO DE UNA RED

El modelamiento de la red también debería ocurrir en una etapa de diseño. El objetivo será definir el procedimiento de implementación de la red, estableciendo qué servicios web construir y cómo se relacionarán unos con otros (dependencias).

Los objetivos de diseño son claros: definir una estrategia de implementación sencilla y segura (que no provoque dificultades ni riesgos), construir una red que no produzca sobrecarga (en especial a terceros), planear dentro del presupuesto, lograr que los servicios web a crear sean rápidos y respondan con prontitud, etc. En este sentido, la evaluación de las métricas de la red puede servir para evaluar aspectos del diseño.

MÉTRICAS A USAR

No se pueden usar métricas de funcionamiento... Al menos, no todas se pueden estimar. Por supuesto, el nivel de servicio es una de esas métricas no estimables. De todas maneras, se puede inspeccionar el diseño de la red para buscar posibles problemas.

Las métricas estructurales se pueden usar perfectamente con este tipo de red. En cuanto a las métricas de funcionamiento, se pueden estimar las métricas de demanda y del uso de las dependencias, y las métricas derivadas. Las métricas de las dependencias (NI y Pr) se pueden estimar según la estructura planeada de implementación. Para estimar la demanda externa, se debe hacer una estimación de la demanda con algún método de pronóstico. El resto de las métricas de demanda se pueden estimar según el procedimiento explicado en la siguiente subsección.

De todas formas, a partir de las métricas Pr, NI y DdaExterna, se pueden estimar:

1. $Nlr(l) = \frac{NI(l)}{Pr(l)} - 1$ (la potencial redundancia en llamadas).
2. $DdaInterna(w)$ (la demanda producida al interior de la red).
3. $Demanda(w) = DdaExterna(w) + DdaInterna(w)$ (la demanda total por servicio).
4. $CargaRGen(w) = \sum_{(w,v) \in L} NI(w,v) \times (1 + CargaRGen(v))$ (la carga real generada).

OBTENCIÓN DE LA DEMANDA INTERNA O DEPENDIENTE

Si se ha hecho sólo la estimación de la demanda externa, la demanda interna es relativamente fácil de estimar. De hecho, se puede determinar con cualquier aplicación que permita resolver sistemas lineales.

La deducción del sistema lineal es:

1. La demanda total de un servicio es la suma de las demandas externa e interna.
2. La demanda externa del servicio w es: $\sum_{(v,w) \in L} Demanda(v) \times Nl(v,w)$
3. La demanda total de w es: $DdaExterna(w) + \sum_{(v,w) \in L} Demanda(v) \times Nl(v,w)$

La fórmula anterior define el siguiente sistema lineal:

$$Demanda(w) = DdaExterna(w) + \sum_{(v,w) \in L} Demanda(v) \times Nl(v,w)$$

En la fórmula anterior, $DdaExterna$ y Nl son datos. Esperando eficiencia, se podría asumir $Nl = Pr$. Más aún, si la dependencia es determinista (siempre se va a ocupar), se puede asumir $Nl = 1$.

Reordenando los términos:

$$Demanda(w) - \sum_{(v,w) \in L} Demanda(v) \times Nl(v,w) = DdaExterna(w)$$

En forma de matriz:

$$\begin{pmatrix} 1 & -Nl(w^2, w^1) & -Nl(w^3, w^1) & \dots & & \\ -Nl(w^1, w^2) & 1 & -Nl(w^3, w^2) & \dots & & \\ -Nl(w^1, w^3) & -Nl(w^2, w^1) & 1 & \dots & & \\ \vdots & & & 1 & & \\ \vdots & & & & 1 & \\ \vdots & & & & & 1 \end{pmatrix} \begin{pmatrix} Demanda(w^1) \\ Demanda(w^2) \\ Demanda(w^3) \\ \vdots \\ \vdots \\ Demanda(w^{Nro(W)}) \end{pmatrix} = \begin{pmatrix} DdaExterna(w^1) \\ DdaExterna(w^2) \\ DdaExterna(w^3) \\ \vdots \\ \vdots \\ DdaExterna(w^{Nro(W)}) \end{pmatrix}$$

Si se asume una red cuyas llamadas ocurren de forma determinista, entonces se puede

definir Nl como: $Nl(l) = \begin{pmatrix} 1 & \text{si } l \in L \\ 0 & \sim \end{pmatrix}$.

$$\begin{pmatrix} 1 & 0 & -1 & \dots \\ -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ \vdots & & & 1 & -1 \\ & & & 0 & 1 \end{pmatrix} \begin{pmatrix} Demanda(w^1) \\ Demanda(w^2) \\ Demanda(w^3) \\ \vdots \\ Demanda(w^{Nro(W)}) \end{pmatrix} = \begin{pmatrix} DdaExterna(w^1) \\ DdaExterna(w^2) \\ DdaExterna(w^3) \\ \vdots \\ DdaExterna(w^{Nro(W)}) \end{pmatrix}$$

La matriz de la izquierda es una matriz repleta de -1 y 0, salvo en la diagonal, donde sólo hay unos. El supuesto determinista simplifica mucho la especificación del sistema lineal.

5.6 Utilización de las métricas

5.6.1 Estrategias para mejorar el nivel de servicio

Es difícil disociar todo lo que es la identificación de un problema a lo que son las acciones a realizar para solucionarlo. Por lo tanto, destinar un pequeño espacio a lo que serían las estrategias de solución -en este escenario- parece conveniente.

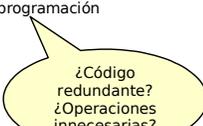
Estrategias genéricas para lograr eficiencia		
Enfocarse en la plataforma	Enfocarse en el servicio web	Enfocarse en la red
Invertir en hardware 	Revisar prácticas ineficientes de programación 	Integración en la red 
Mejorar el sistema operativo 	Análisis del algoritmo 	Dividir la carga (colocar servicios web alternativos) 

Figura 23: Estrategias genéricas para el nivel de servicio.

En la figura 23 se proponen algunas estrategias genéricas. Se han dividido en tres aspectos básicos: la plataforma, el servicio web y la red. A continuación se presenta un resumen conciso de las estrategias:

1. *Enfocarse en la plataforma.* Esta estrategia consiste en alterar elementos del servidor para lograr una mejor calidad de servicio.

1. Invertir en hardware: mejorar el rendimiento a través del cambio de algún componente de un máquina, o toda la máquina.

2. Mejorar el sistema operativo: o cualquier software de soporte de los servicios web.

2. *Enfocarse en el servicio web.* Esta estrategia consiste en mejorar la propia implementación del servicio web con problemas.

1. Revisar prácticas ineficientes de programación: sacar los vicios de código del servicio web para esperar una mejora en el rendimiento.

2. Análisis del algoritmo: tratar de implementar un algoritmo óptimo para el corto intervalo de tiempo en el que debe responder un servicio web.

3. *Enfocarse en la red.* Esta estrategia considera elementos de varios servicios web -pasando al dominio de la red de servicios- para mejorar la calidad de servicio.

1. Integración de la red: implementar funcionalidades de algunas de las dependencias directas en el propio servicio web, para reducir dependencias o destruir dependencias mixtas.

2. Dividir la carga: montar un servidor alternativo o alejar servicios web alternativos en otros servidores para reducir la demanda individual de los servicios.

5.6.2 Evaluación del impacto de una mejora

BENEFICIO SOCIAL DEBIDO A UNA REDUCCIÓN DEL TIEMPO DE SERVICIO

Volviendo a la función de provisión de un servicio electrónico, el costo social del servicio es:

$$\begin{bmatrix} \text{Costo} \\ \text{Social} \end{bmatrix} = \begin{bmatrix} \text{Costo} \\ \text{Ciudadano} \end{bmatrix} \times \begin{bmatrix} \text{Demanda} \\ \text{Servicio} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Provisión} \end{bmatrix}$$

En la fórmula anterior, el costo al ciudadano corresponde a:

$$\begin{bmatrix} \text{Costo} \\ \text{Ciudadano} \end{bmatrix} = \begin{bmatrix} \text{Acceso} \\ \text{Internet} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Tiempo} \end{bmatrix} \times \begin{bmatrix} \text{Comprensión} \\ \text{Interfaz} \end{bmatrix} + \begin{bmatrix} \text{Costo} \\ \text{Tiempo} \end{bmatrix} \times \begin{bmatrix} \text{Tiempo} \\ \text{Respuesta} \end{bmatrix}$$

Presentado esto, una reducción del tiempo de servicio disminuye el costo social, o sea, se convierte en un beneficio social (es un ahorro de un costo social). Considerando a ΔT como la reducción del tiempo de servicio, el valor socio-económico de ese beneficio corresponde a:

$$\text{Beneficio} = -\Delta \begin{bmatrix} \text{Costo} \\ \text{Social} \end{bmatrix} = \begin{bmatrix} \text{Demanda} \\ \text{Servicio} \end{bmatrix} \times \begin{bmatrix} \text{Costo} \\ \text{Tiempo} \end{bmatrix} \times \Delta T$$

A continuación se presenta un par de ejemplos de evaluación socio-económica de beneficios:

1. Se ha mejorado el sólo el servicio web w , ya sea mejorando el código o alterando sus dependencias. En este caso, el beneficio es: $BSocial = \Delta T \times [\text{Costo Tiempo}] \times \text{Demanda}(w)$.

2. Se han mejorado dos o más servicios web por separado. Este caso es complicado. Si se usa la fórmula anterior de manera directa en una mejora simultánea, se estarían contando varias veces los beneficios. Por eso, es necesario actuar secuencialmente para evaluar los beneficios. La estrategia recomendada es realizar una mejora, evaluarla, realizar otra mejora, evaluarla, etc. así sucesivamente. Al seguir una estrategia secuencial, se está permitiendo que las métricas se actualicen. Así, se puede volver a usar la fórmula anterior.

3. Se han mejorado varios servicios web de forma simultánea. Si el cambio ha sido simultáneo, es necesario calcular el ahorro de tiempo para cada servicio web con demanda externa, sin importar cuánto hayan cambiado internamente las dependencias (mientras se mantengan los mismos servicios). La nueva fórmula para evaluar los beneficios sociales es: $Bsocial = \sum_{w \in W} (\Delta T^w \times DdaExterna(w) \times [\text{Costo Tiempo}])$, siendo ΔT^w la reducción del tiempo en el servicio web w . Al considerar sólo la demanda externa, se está evitando contar dos veces; los servicios que son pedidos por la demanda externa han absorbido todos los beneficios de ahorro de tiempo de la red.

¿Cómo obtener el ahorro de tiempo en un servicio web? A través de la métrica que mide el tiempo, $T_{\text{Respuesta}}$. De hecho: $\Delta T^w = \Delta T_{\text{Respuesta}}(w) = T_{\text{Respuesta}}|_{t_1}(w) - T_{\text{Respuesta}}|_{t_0}(w)$. Es importante dejar registro histórico de las métricas para realizar una evaluación correcta.

La fórmula general para medir los beneficios de la red debido a un ahorro de tiempo, sin que cambien los servicios web, es:

$$B_{\text{social}} = \sum_{w \in W} (\Delta T^w \times Dda_{\text{Externa}}(w) \times [\text{Costo Tiempo}])$$

El costo social del tiempo es algo que debe ser evaluado según el tipo de usuarios de los servicios. No todos los usuarios son iguales. Luego, para lograr una mejor estimación de los beneficios sociales, es necesario considerar el costo social del tiempo según el perfil del ciudadano. Para un costo social del tiempo de un ciudadano cualquiera, consulte el apéndice B.

Para una evaluación socio-económica real, es necesario ir más allá del beneficio descubierto en el corto plazo por las fórmulas anteriores. Los elementos a considerar en la evaluación dependerán de cada caso en particular.

Ejemplo

Un organismo ha contratado un par de programadores para mejorar su red. El resultado fue que dos de sus servicios web redujeron sus tiempos de respuesta en sólo dos segundos. Uno de sus servicios tiene una demanda externa de 5 mil accesos mensuales y el otro de 75 mil accesos mensuales. Si la demanda crece un 5% anual, la tasa social de descuento es del 8% anual, el costo social del tiempo es de \$1200/hora y el costo de la mejora fue de dos millones de pesos⁴⁹ (pago de los programadores), el valor del proyecto es de casi \$11 millones para un horizonte de diez años (figura 24). La razón IVAN es de 5,49.

49 Se considera porque se está evaluando el proyecto como un todo. De lo contrario, esos dos millones de pesos se habrían considerado un costo hundido.

Años	0	1	2	3	4	5	6	7	8	9	10
Servicio 1		\$ 40000	\$ 42000	\$ 44100	\$ 46305	\$ 48620	\$ 51051	\$ 53604	\$ 56284	\$ 59098	\$ 62053
Servicio 2		\$ 600000	\$ 630000	\$ 661500	\$ 694575	\$ 729304	\$ 765769	\$ 804057	\$ 844260	\$ 886473	\$ 930797
Resultado	-2000000	\$ 640000	\$ 672000	\$ 705600	\$ 740880	\$ 777924	\$ 816820	\$ 857661	\$ 900544	\$ 945571	\$ 992850
VPN	\$ 10981465										

Figura 24: Evaluación socio-económica (ejemplo).

El flujo de caja presentado en la figura 24 es bastante sencillo. Ello se debe a que las mejoras se evalúan como *proyectos de extensión* (son modificaciones a proyectos existentes) y a que la complejidad está fuera del flujo de caja.

IMPACTO DEBIDO AL MENOR USO DE LA RED

La evaluación socio-económica de este aspecto es más complicada. Sin embargo, se puede evaluar a través de la métrica *CargaRGen* (carga real generada). Para evaluar este aspecto de los proyectos de mejora, se hace conveniente usar una razón costo-impacto, como $CargaRGen/Costo$. Así, se pueden comparar proyectos similares y acumular aprendizaje al respecto.

5.7 Resumen

En este capítulo se ha desarrollado un modelo completo de análisis de las redes de servicios web en el gobierno electrónico, pasando tanto por fundamentos económicos como por estrategias de desarrollo de software.

Al principio del capítulo se presentan dos observaciones sobre la economía del servicio electrónico: la función de *provisión* del servicio y los efectos económicos de estar en red. A partir de las observaciones de los efectos económicos de la red se desarrolla un modelo basado en dependencias entre servicios web. El modelo corresponde a un grafo dirigido cuyos nodos son los servicios web presentes en una red y sus arcos son las “llamadas” entre los servicios.

A partir de la presentación de potenciales problemas en la red, se inicia el desarrollo de las métricas del modelo. Las métricas miden aspectos de funcionamiento de los servicios web (métricas de funcionamiento, basadas en estimadores estadísticos) y elementos de la estructura de la red (métricas estructurales). La definición de las métricas es muy concreta y no se presta a

ambigüedades.

Luego se presenta cómo modelar una red real y una red hipotética (una red en etapa de diseño). Se explican fuentes de datos y otros aspectos a tener en consideración.

Finalmente, se presentan un pequeño modelo con las posibles acciones a realizar para mejorar el nivel de servicio de la red (modelo de estrategias genéricas) y cómo evaluar el impacto de las mejoras. La evaluación de impacto está altamente basada en los conceptos económicos tras la función de provisión del servicio electrónico. Se muestra cómo evaluar socio-económicamente una mejora en el tiempo de respuesta y cómo evaluar el impacto de una menor carga en la red.

6 Extensiones al modelo

6.1 Introducción

El modelo presentado en el capítulo anterior -que básicamente consiste en una metodología de modelamiento con un modelo para el análisis- se puede extender para considerar los siguientes casos:

1. UN MODELO CON CICLOS. Si bien la existencia de ciclos puede indicar un mal uso de la *extensibilidad* de una red SOA (la que se extiende mediante la incorporación de nuevos servicios), los ciclos pueden aparecer en la práctica. Además, una red sin ciclos se puede ver como un caso particular de un modelo que admite ciclos.
2. UN MODELO DE GRUPOS DE SERVICIOS WEB. En el modelo presentado, los servicios web correspondían a los nodos y las llamadas entre estos servicios correspondían a los arcos. Esto requiere información detallada para el modelamiento. Sin embargo, es posible hacer que los nodos representen a grupos de servicios web, permitiendo un apoyo más general a la hora de hacer diseños.
3. LA INCLUSIÓN DE OTROS ELEMENTOS A LA RED. El modelo está pensado para redes de servicios web. Sin embargo, es posible describir cualquier arquitectura SOA dentro del modelo (aunque el sentido de las métricas puede variar). De todas maneras, cualquier funcionalidad que opere como un servicio en una arquitectura SOA se puede incluir en el modelo.

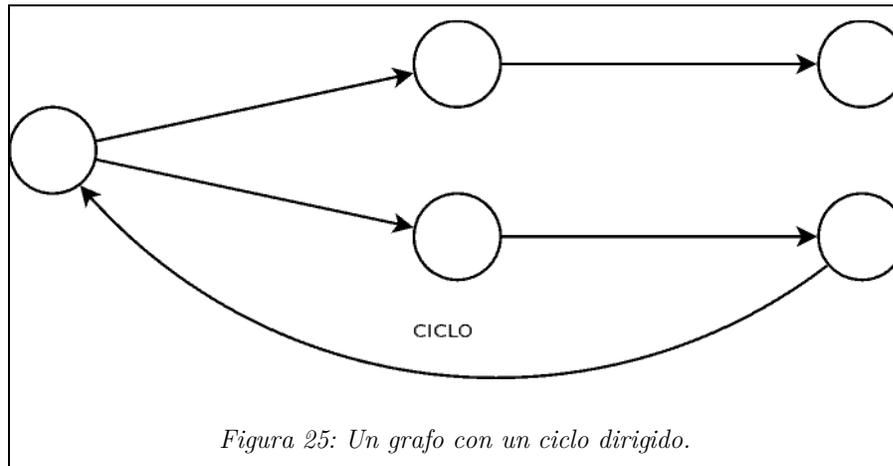
En este capítulo se muestra cómo estas extensiones alteran el modelo original.

6.2 Extensión: Tolerancia a ciclos

6.2.1 El problema

No todas las redes estarán carentes de ciclos⁵⁰. Probablemente haya un buen número de ciclos en las redes existentes. Si bien los ciclos no son recomendables por una cuestión de sanidad del desarrollo y de control de la carga de la red, perfectamente son admisibles en una arquitectura SOA.

⁵⁰ Ciclos dirigidos.



¿Qué problema hay con los ciclos en una red con ciclos dirigidos? Los ciclos debilitan conceptos como el grado de conectividad, la concentración y la desconcentración de una red. Y otros conceptos desaparecen, como la carga estructural generada (CargaSGen), que, de hecho, no se puede calcular.

Cabe destacar que no se está modelando un *hipergrafo*⁵¹: entre dos nodos no puede haber más de un arco. Tampoco es posible para un servicio web llamarse a sí mismo (es un grave error de programación).

6.2.2 Cambios en las métricas

A continuación se presentan los cambios, tanto en cálculo como en sentido, que tienen las métricas presentadas en el capítulo anterior:

1. *NServicio*. Esta métrica no mide nada directamente relacionado a la red. Por esto, no cambia en absoluto.
2. *NServS*. Esta métrica no mide nada directamente relacionado a la red. Por esto, no cambia en absoluto.
3. *Entrantes*. Sigue siendo válida. No pierde sentido.
4. *Salientes*. Sigue siendo válida. No pierde sentido.
5. ~~*CargaSGen*~~. Si hubieran ciclos en la red, esta métrica se vuelve incalculable. **Ya no**

⁵¹ Un hipergrafo es una extensión de un grafo que admite más de un arco entre dos nodos.

es válida.

6. *GrConex*. Su sentido permanece sin cambios. Sin embargo, es una razón entre el número de arcos en el grafo y la cantidad máxima de arcos en un grafo dirigido acíclico. Pero considerando nocivos a los ciclos, se deja intacta para comparar contra una red sin ciclos.

7. *GrConcen*. La concentración de la red es un concepto relacionado a las dependencias directas (llamadas). Y según este modelo, el ciclo más corto es una dependencia indirecta (el camino más corto sería algo como $c=(a, b, a)$), por lo que todo ciclo, al representar caminos con 3 o más elementos, es una dependencia indirecta. Luego, la definición de *GrConcen* no lidia con ciclos.

8. *GrDesconcen*. Permanece válida, al igual que *GrConcen*.

9. *CargaSub*. La carga mínima que debería requerir un servicio web, en términos de servicios a llamar. Los ciclos podrían implicar necesitar de un mismo servicio infinitas veces, pero al considerarse un único uso por ejecución, esta métrica mantiene su sentido. Como su definición era conjuntista, no necesita ser alterada.

10. ~~*ReCarga*~~. Al ser *CargaSGen* inválida, *ReCarga* también lo es (por fórmula).

11. *Pr*. Es una propiedad de un arco, independiente del resto de la red. Sigue siendo válida.

12. *Nl*. Es una propiedad de un arco, independiente del resto de la red. Sigue siendo válida.

13. *Nlr*. Es una propiedad de un arco, independiente del resto de la red. Sigue siendo válida.

14. *TRespuesta*. Es una propiedad de un nodo, independiente del resto de la red. Sigue siendo válida.

15. *VRespuesta*. Es una propiedad de un nodo, independiente del resto de la red. Sigue siendo válida.

16. *PRespuesta*. Depende de los vecinos, a partir de *TRespuesta*. Al ser esta última válida, entonces *PRespuesta* también lo es.

17. *Demanda*. Es una propiedad de un nodo, independiente del resto de la red. Sigue siendo válida.

18. *DdaExterna*. Depende de los vecinos, a partir de *Demanda*. Al ser esta última válida, entonces *DdaExterna* también lo es.

19. *DdaInterna*. Al depender de *Demanda* y *DdaExterna*, ambas válidas, *DdaInterna* también lo es.

20. *CargaRGen*. La métrica original no consideraba el posible “eco” de carga que podría recibir un servicio web al estar en un ciclo. Para medir todos los efectos de carga real generada, incluso sobre el mismo servicio web, es necesario calcular de forma diferente esta métrica.

○ Fórmula original:
$$CargaRGen(w) = \sum_{(w,v) \in L} Nl(w,v) \times (1 + CargaRGen(v))$$

○ El problema de la fórmula anterior es su recursividad. Si hay ciclos, una función recursiva *inocente* terminaría dependiendo de su propio valor: $g(x) = f(g(x))$.

○ Más apropiado es observar que la fórmula define un sistema lineal. Si la solución se obtiene de resolver el sistema lineal, se evita el problema de una implementación recursiva.

○ Para simplificar el cálculo, sea $AuxRGen(w) = 1 + CargaRGen(w)$.

○ La fórmula original de *CargaRGen* se puede escribir en función de *AuxRGen*, simplificándose:
$$AuxRGen(w) - 1 = \sum_{(w,v) \in L} Nl(w,v) \times AuxRGen(v)$$

○ Ordenando:
$$AuxRGen(w) - \sum_{(w,v) \in L} Nl(w,v) \times AuxRGen(v) = 1$$

○ Finalmente, la ecuación anterior se puede escribir en forma matricial⁵² como:

52 El sistema **debe** tener solución. De lo contrario, la red se estaría saturando a sí misma debido a las llamadas.

$$\begin{pmatrix} 1 & -Nl(w^2, w^1) & -Nl(w^3, w^1) & \dots & \\ -Nl(w^1, w^2) & 1 & -Nl(w^3, w^2) & \dots & \\ -Nl(w^1, w^3) & -Nl(w^2, w^1) & 1 & \dots & \\ \vdots & & & 1 & \\ \vdots & & & & 1 \end{pmatrix} \begin{pmatrix} AuxRGen(w^1) \\ AuxRGen(w^2) \\ AuxRGen(w^3) \\ \vdots \\ AuxRGen(w^{Nro(W)}) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

○ Luego de obtener *AuxRGen*, basta sustraer la unidad para obtener *CargaRGen*.

21. *PrDisponible*. Es una propiedad de un nodo, independiente del resto de la red. Sigue siendo válida.

6.3 Extensión: red de grupos

6.3.1 Definiciones

Los servicios web pueden mutar; así como es posible combinar varios servicios web en uno, es posible dividir un servicio web en varios otros. Partiendo del detalle, de los servicios web y sus interconexiones, ¿es posible agregar información a través de agrupar nodos? Y pensando en arquitecturas mayores, ¿es posible modelar grandes bloques de sistemas informáticos? Esta pregunta lleva al modelamiento de grupos de servicios web y al análisis de la interoperabilidad entre instituciones.

Definición. *Una red de grupos es un grafo dirigido en el que sus nodos son conjuntos de servicios web y sus vértices son dependencias entre estos nodos.*

El objetivo tras la red de grupos es otorgar una nueva dimensión de modelamiento, para un análisis de carácter global, más simple y menos preciso. Además, al definir los nodos como “conjuntos de servicios web”, no es necesario modelar en detalle o conocer todos los servicios web de una red. Esto permite modelar redes que no se conocen en detalle o que aún no se construyen (etapa de diseño). La figura 26 permite apreciar el concepto tras la red de grupos.

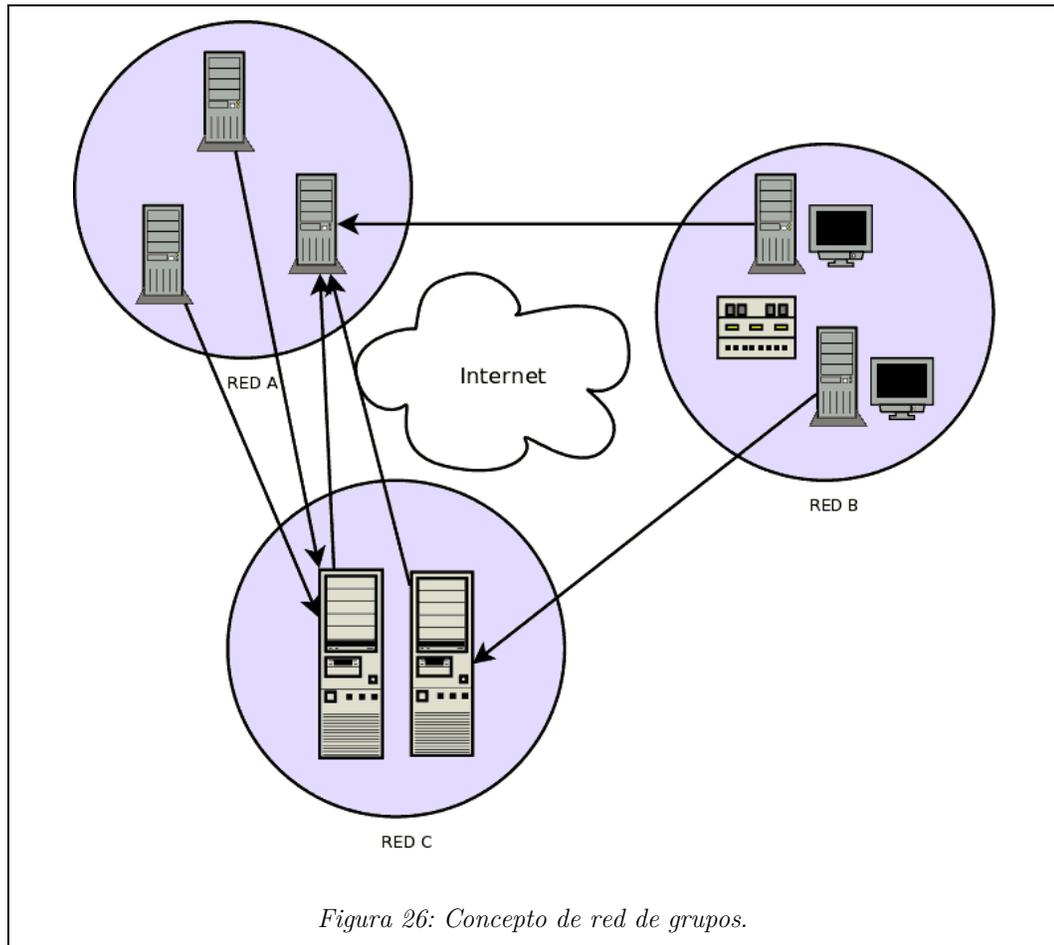


Figura 26: Concepto de red de grupos.

Definición. Una red de grupos es un grafo dirigido $G(N, A)$, siendo N el conjunto de grupos de servicios web ($N : n \in N \Rightarrow n \subseteq W$) y $A \subseteq N \times N$ el conjunto de dependencias entre grupos.

El concepto de dependencia, ahora, hace referencia a la necesidad de cooperación entre un par de grupos de servicios web (interoperabilidad). También se puede ver como la necesidad de información entre grupos.

6.3.2 Conversión de una red de servicios web a una red de grupos

Si se posee una red de dependencias de servicios web, $G(W, L)$, entonces es posible construir una red de grupos a partir de ésta.

Sea $g(W, L)$ una red de dependencias de servicios web, con W el conjunto de servicios web y L el conjunto de llamadas (dependencias directas). Ahora, el objetivo será construir la red

$G(N, A)$, donde N es un conjunto de grupos de servicios web y A son las llamadas entre estos grupos.

Un primer paso es definir los grupos, o sea, los elementos de N . Como se indicó, se debe cumplir: $\forall n \in N : n \subseteq W$. Los servicios web debieran ser agrupados de acuerdo a una característica común, por ejemplo, por organismo, unidad, ubicación, etc. Además, un servicio web debiera quedar considerado dentro de no más de un grupo. Esto hace que los grupos sean disjuntos: $\forall n, m \in N : n \neq m \Rightarrow n \cap m = \emptyset$. La figura 27 ejemplifica una red de grupos sobre una red de servicios web.

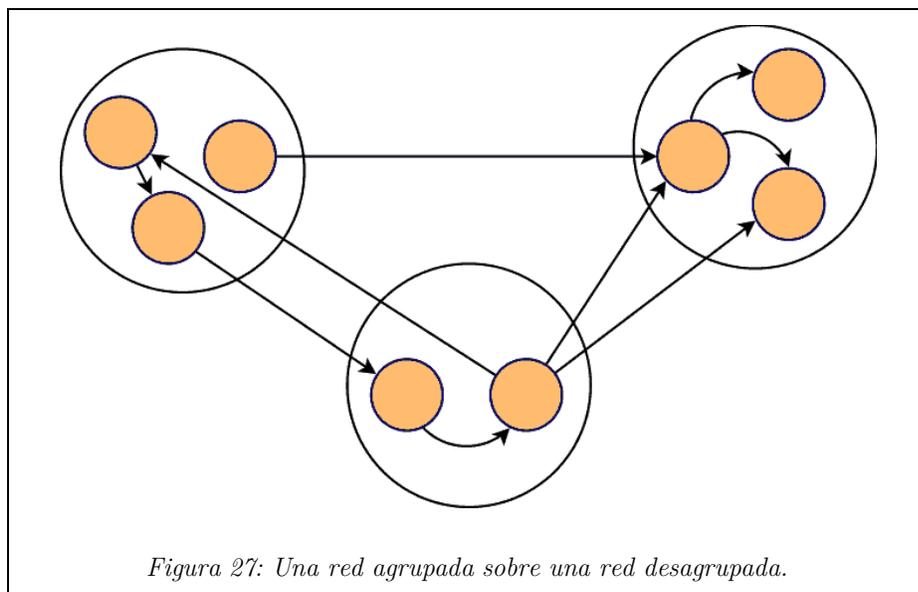


Figura 27: Una red agrupada sobre una red desagrupada.

Luego, es necesario convertir algunas métricas básicas como Pr, Nl y Demanda. Primero, es necesario calcular la demanda del grupo:

```

Para cada grupo B en N:
  Construir L':
     $L' = L \cap B \times B$ 

  Construir la red de dependencias:  $g'(B, L')$ 
  Calcular DdaExterna en  $g'$ 
  Calcular la Demanda de B:
     $Demanda(B) = \sum_{w \in B} DdaExterna(w)$ 
    
```

Código 3: Cálculo de la demanda de los grupos.

Establecida la demanda, se pueden calcular Pr y Nl:

Para cada par de grupos A,B en N:
Calcular Pr mediante:

$$Pr(A, B) = 1 - \prod_{\substack{(a, b) \in L: \\ a \in A, b \in B}} (1 - Pr(a, b))^{\frac{Dda(a)}{Dda(A)}}$$

Calcular Nl mediante:

$$Nl(A, B) = \sum_{\substack{(a, b) \in L \\ a \in A, b \in B}} \left(\frac{Dda(a)}{Dda(A)} \times Nl(a, b) \right)$$

Código 4: Cálculo de Pr y Nl en la red de grupos.

Todas las métricas que se basan en Nl, Pr y Demanda pueden ser calculadas. Sin embargo, su sentido deja de ser tan importante a causa de la falta de precisión que sale de agrupar la información. De todas maneras, el uso de CargaRGen puede ser bastante interesante, ya que mide la carga entre grupos.

6.3.3 Construcción de una red de grupos como un diseño

El diseño de una red de grupos puede ayudar a delinear *a grosso modo* el diseño de una red de servicios web.

A la hora de diseñar una red de grupos, es muy importante definir coherentemente qué son los nodos: si son redes de computadores, instituciones, etc. La coherencia es parte de un buen diseño. Debe tenerse en cuenta que cada grupo debe poseer alguna función o información que el resto de los grupos no posea o no otorgue. Si un grupo no se distingue de otro, ese grupo sería redundante.

Las dependencias debieran modelarse según las necesidades de información o de funcionalidad entre los grupos. Por ejemplo, Aduana⁵³ necesita información de Registro Civil para identificar vehículos. Entonces, si un nodo de la red corresponde a Aduana (A) y otro corresponde a Registro Civil (Rc), entonces la dependencia (A,Rc) debe estar presente en el modelo.

La demanda externa de los grupos se puede determinar a partir de información real sin muchos problemas, si esa demanda corresponde a algo existente. De lo contrario, sería necesario

⁵³ El caso de Aduana se trata de manera especial en el capítulo .

realizar algún estudio para realizar la estimación. Una vez pronosticada la demanda externa, la demanda total de cada nodo se puede obtener con el método presentado en la sección 5.5.2.

6.3.4 Métricas aplicables a la red de grupos

Las métricas orientadas a la precisión y a los servicios web como tales ya no son aplicables. Tampoco se aplican aquellas métricas inválidas para las redes con ciclos.

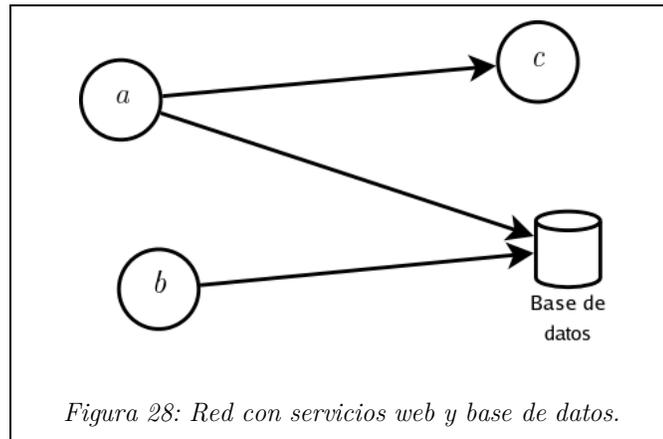
22. ~~*NServicio*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.
23. ~~*NServS*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.
24. *Entrantes*. Sirve para observar cuántos grupos dependen directamente de otro.
25. *Salientes*. Sirve para observar de cuántos grupos depende directamente uno.
26. ~~*CargaSGen*~~. Inválida debido a la posible presencia de ciclos.
27. *GrConex*. Pierde sentido a medida que los grupos de nodos se hacen más grandes.
28. *GrConcen*. Permanece válida, aunque su sentido depende de lo que se esté modelando.
29. *GrDesconcen*. Permanece válida, al igual que *GrConcen*.
30. ~~*CargaSub*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.
31. ~~*ReCarga*~~. Inválida por la presencia de ciclos.
32. *Pr*. Es una propiedad de un arco, independiente del resto de la red. Sigue siendo válida.
33. *Nl*. Es una propiedad de un arco, independiente del resto de la red. Sigue siendo válida.
34. *Nlr*. Es una propiedad de un arco, independiente del resto de la red. Sigue siendo válida.

35. ~~*TRespuesta*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.
36. ~~*VRespuesta*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.
37. ~~*PRespuesta*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.
38. *Demanda*. Es una propiedad de un nodo, independiente del resto de la red. Sigue siendo válida. Pasa a ser la demanda del grupo.
39. *DdaExterna*. Depende de los vecinos, a partir de *Demanda*. Al ser esta última válida, entonces *DdaExterna* también lo es.
40. *DdaInterna*. Al depender de *Demanda* y *DdaExterna*, ambas válidas, *DdaInterna* también lo es.
41. *CargaRGen*. Sigue siendo válida, y su método de cálculo corresponde al del caso con ciclos (que es el caso general). Su significado varía según los grupos que se están modelando (tamaño de los grupos y criterio de agrupación).
42. ~~*PrDisponible*~~. No tiene sentido. No se está modelando un servicio web o aplicación similar.

6.4 Extensión: Nodos de distinta naturaleza

¿Cuánto de SOAP se está modelando en la red? Sólo el aspecto arquitectónico: SOA.
¿Cuánto de XML se está modelando en la red? Nada. Estas observaciones son prácticas para justificar esta extensión: el modelo admite muchos tipos de redes SOA, no sólo redes de servicios web.

(Ejemplo) En la figura 28, los servicios web *a* y *b* se conectan a una base de datos remota a través de Internet. La base de datos ofrece un servicio de consultas de información. Naturalmente, *a* y *b* dependen de la base de datos. Si se cae la base de datos, *a* y *b* también.



El ejemplo anterior ilustra cómo un elemento diferente a un servicio web también es admitido por los principios del modelo de dependencias. Además, una base de datos también es un servicio. Así, cualquier funcionalidad similar puede ser incluida en el modelo.

6.5 Comentarios

En este capítulo se han presentado y discutido tres extensiones al modelo presentado en el capítulo anterior:

1. *La inclusión de ciclos dirigidos.* Algunas métricas se invalidan mientras otras cambian su fórmula de obtención.
2. *El modelamiento de grupos de servicios web.* Tomando base en la extensión con ciclos dirigidos, se presenta una extensión que permite modelar la interacción de grupos de servicios web. Muchas métricas dejan de tener sentido en este caso.
3. *El modelamiento de otros elementos que prestan servicios, además de los servicios web.* Debido a la flexibilidad del modelo original, es posible modelar arquitecturas SOA de varios tipos que interactúan en red. Esto permite mezclar diferentes elementos que prestan servicios, y modelarlos.

La primera extensión se puede ocupar conjuntamente con las otras extensiones. Sin embargo, las dos últimas extensiones son excluyentes.

7 Casos de estudio

7.1 Pruebas teóricas y desarrollo

7.1.1 Aplicación desarrollada

En base al modelo elaborado para evaluar redes de servicios web (el modelo de red de dependencias y las métricas asociadas), se desarrolló una aplicación web que permite visualizar la red y calcular algunas métricas según la información suministrada.

La aplicación web fue desarrollada en lenguaje PHP y los gráficos son presentados en formato SVG (Scalable Vector Graphics). Se ha probado en un equipo con un servidor Apache versión 2.2, intérprete PHP versión 5.1.2 y sistema operativo Fedora Core 5, y en un equipo con servidor Apache 2.0.58, intérprete PHP 4.4.4-pl6-gentoo y sistema operativo Gentoo con kernel 2.6. La visualización de la aplicación web ha sido probada con Mozilla Firefox 1.5.0.1 (en Windows y en Linux) y Konqueror 3.5.1-5 (de KDE 3.5.1-2.3 Red Hat). La visualización de los gráficos SVG no ha sido soportada por Internet Explorer.

El ingreso de la red de dependencias se hace a través de formato XML. El lenguaje usado es propio debido a que los actuales lenguajes de grafos no permiten el ingreso de variables y otros atributos tanto a arcos como a nodos.

7.1.2 Utilización de la aplicación

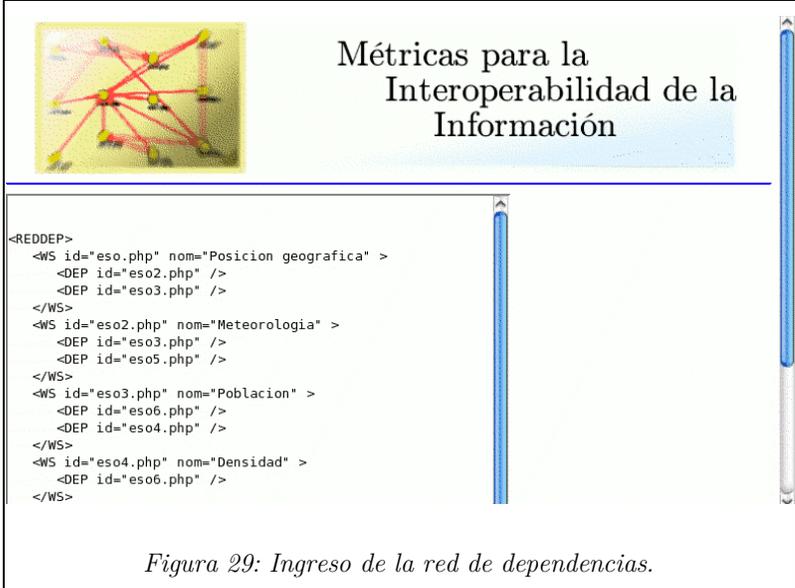
El procedimiento para trabajar con la aplicación desarrollada es el siguiente:

1. Ingreso de la red de dependencias.
2. Cálculo de las métricas necesarias.
3. Visualización de los resultados.

Para permitir un flujo de trabajo más flexible, el primer paso es obligatorio sólo al comenzar el uso de la aplicación. Luego, todas las acciones se pueden realizar en cualquier orden. Naturalmente, ingresar una nueva red de dependencias elimina la información anterior de las métricas.

La primera pantalla de la aplicación consiste en una caja de texto para ingresar la red de

dependencias en formato XML. Esto se aprecia en la figura 29.



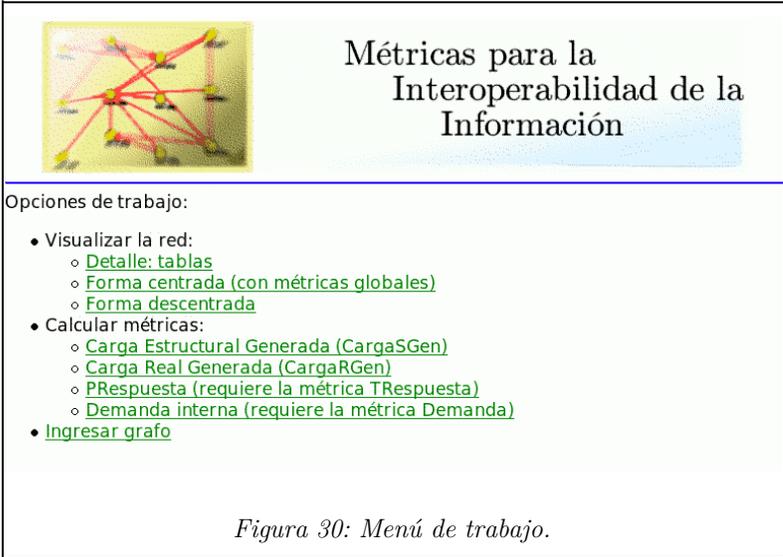
Métricas para la Interoperabilidad de la Información

```

<REDDEP>
<WS id="eso.php" nom="Posicion geografica" >
<DEP id="eso2.php" />
<DEP id="eso3.php" />
</WS>
<WS id="eso2.php" nom="Meteorologia" >
<DEP id="eso3.php" />
<DEP id="eso5.php" />
</WS>
<WS id="eso3.php" nom="Poblacion" >
<DEP id="eso6.php" />
<DEP id="eso4.php" />
</WS>
<WS id="eso4.php" nom="Densidad" >
<DEP id="eso6.php" />
</WS>
    
```

Figura 29: Ingreso de la red de dependencias.

El menú principal de trabajo aparece después de la depuración del código ingresado. En el menú principal (figura 30) se puede visualizar la red y las métricas ya calculadas, se pueden calcular otras métricas más y se puede ingresar una nueva red.



Métricas para la Interoperabilidad de la Información

Opciones de trabajo:

- Visualizar la red:
 - [Detalle: tablas](#)
 - [Forma centrada \(con métricas globales\)](#)
 - [Forma descentrada](#)
- Calcular métricas:
 - [Carga Estructural Generada \(CargaSGen\)](#)
 - [Carga Real Generada \(CargaRGen\)](#)
 - [PRespuesta \(requiere la métrica TRespuesta\)](#)
 - [Demanda interna \(requiere la métrica Demanda\)](#)
- [Ingresar grafo](#)

Figura 30: Menú de trabajo.

Por ejemplo, la opción “forma centrada” dentro de “Visualizar la red” permite ver el grafo de la red de dependencias. Además, esta forma de visualización muestra las métricas estructurales

globales de la red, como se aprecia en la figura 31.

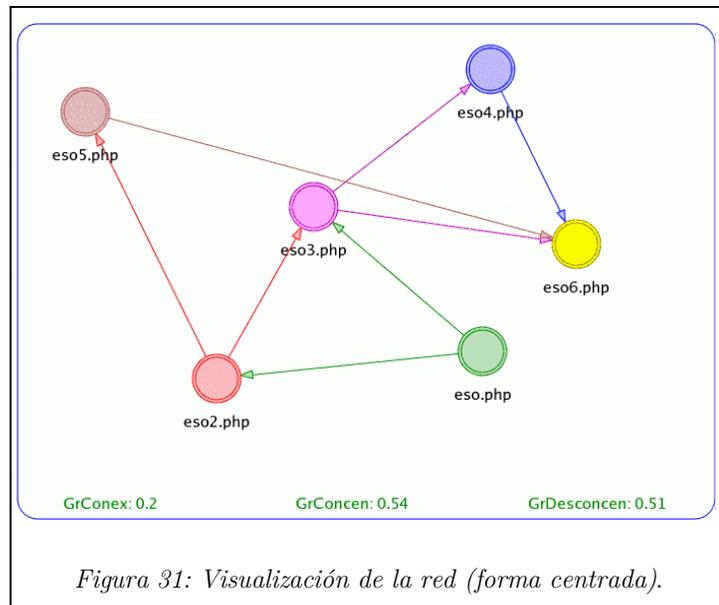


Figura 31: Visualización de la red (forma centrada).

Otras opciones permiten calcular métricas. Por ejemplo, la opción “Carga Estructural Generada” dentro de “Calcular Métricas”, permite calcular CargaSGen y visualizar la métrica en el grafo, como se puede apreciar en la figura 32.

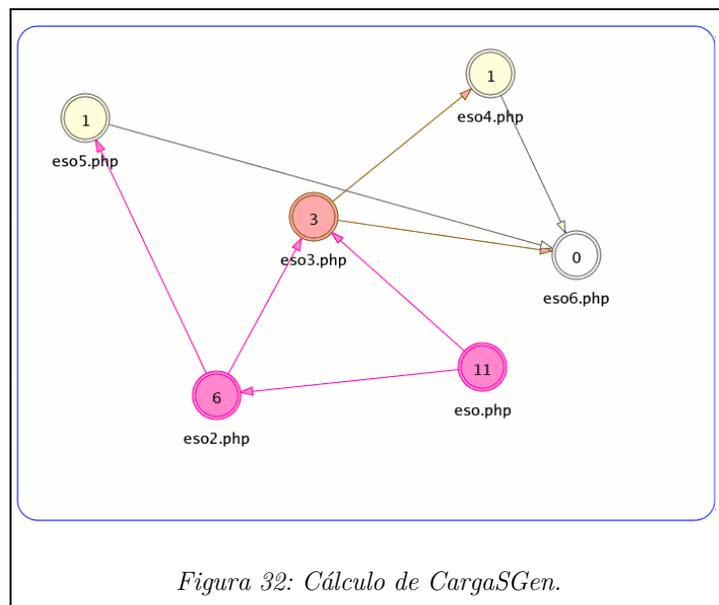


Figura 32: Cálculo de CargaSGen.

Luego de calcular una métrica, es posible ver la información detallada de la red en el link

“Detalle: Tablas”. El estado actual de los cálculos se presenta aquí. Es el reflejo de la información disponible. El formato de visualización corresponde a una tabla por servicio web (figura 33).

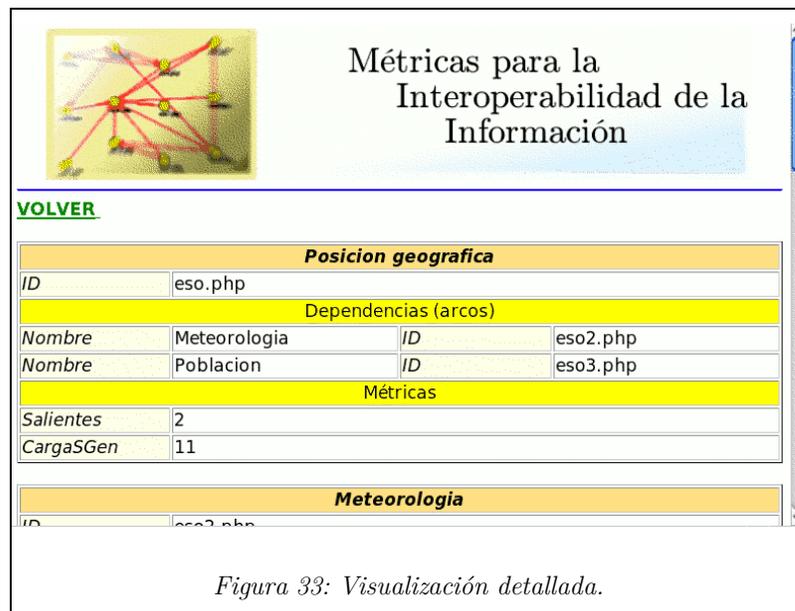


Figura 33: Visualización detallada.

Se ha usado esta aplicación para el modelamiento de las redes.

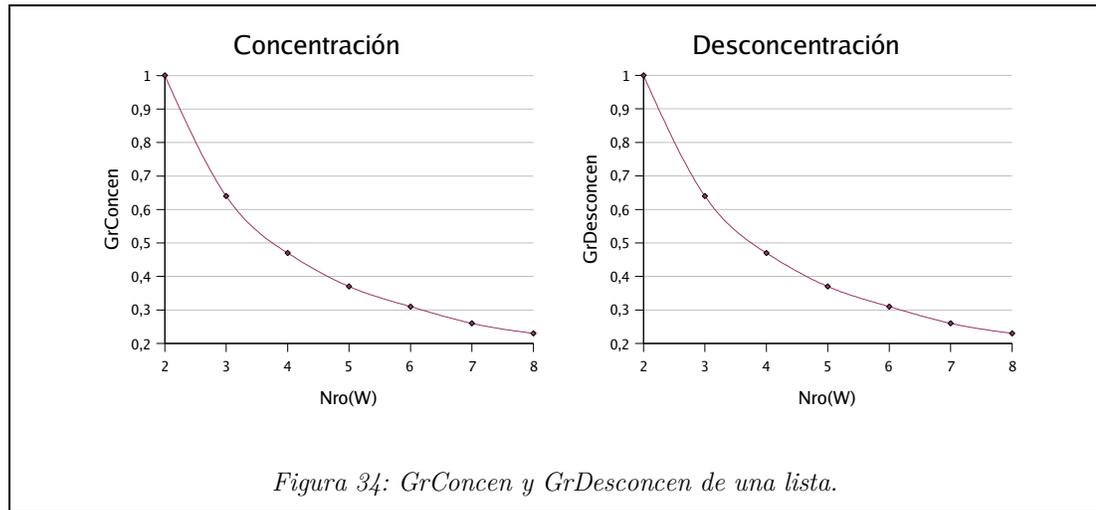
7.1.3 Pruebas teóricas

Estas pruebas teóricas están orientadas a observar el comportamiento de las métricas ante algunas situaciones, con el fin de comprobar si revelan la información deseada. Las pruebas se han hecho sobre las métricas más complicadas de observar.

CONCENTRACIÓN Y DESCONCENTRACIÓN DE LA RED

Cuando muchos nodos dependen de unos pocos nodos en la red, esa red está concentrada en esos nodos. Cuando los nodos dependen de muchos otros, la red está desconcentrada. Estos conceptos pueden parecer intuitivos, pero dejan de serlo frente a redes con forma extraña. Sin embargo, estos conceptos son importantes: hacen alusión a los riesgos de dependencia a nivel de red. Valores muy altos implican un gran riesgo.

La primera serie de experimentos consiste en observar las métricas ante una lista. Una lista es una estructura cuyos nodos están conectados sólo a uno más, salvo por último. El experimento se realizó con 2 a 8 nodos.



La figura 34 ilustra los resultados del experimento. Efectivamente, la lista no es concentrada ni desconectada. Sin embargo, cuando el número de nodos es pequeño, es razonable ver que estos valores son altos. La concentración de la red es una noción válida para números razonablemente grandes de servicios web.

SIMULACIÓN DE UN ANÁLISIS

Supongamos que una red tiene un servicio web A llama a B con $Nl=0,8$ y a C con $Nl=1,5$. El servicio B no llama a ningún otro. El servicio C llama D con $Nl=0,5$ y a E con $Nl=0,6$. Los servicios D y E no llaman a ningún otro. Las demandas son Demanda(A)=50, Demanda(B)=30, Demanda(C)=80, Demanda(D)=200 y Demanda(E)=100 (obtenidas de los logs). ¿Cómo se puede analizar esta red?

Un primer paso consiste en escribir en el formato adecuado. El formato sería:

```
<REDDEP>
  <WS id="A" nom="A" >
    <MED nom="Demanda" val="50" />
    <DEP id="B" nl=".8"/>
    <DEP id="C" nl="1.5"/>
  </WS>
  <WS id="B" nom="B" > <MED nom="Demanda" val="30" /> </WS>
  <WS id="C" nom="C" >
    <MED nom="Demanda" val="80" />
    <DEP id="D" nl=".5"/>
    <DEP id="E" nl=".6"/>
  </WS>
  <WS id="D" nom="D" > <MED nom="Demanda" val="200" /> </WS>
  <WS id="E" nom="E" > <MED nom="Demanda" val="100" /> </WS>
</REDDEP>
```

Código 5: Formato de ingreso de una red.

El formato de ingreso es bastante sencillo (código 5). Para agregar un nuevo servicio web, se escribe `<WS id="id" nom="nombre">`. El nombre es opcional. Dentro de esa etiqueta se agregan las métricas `<MED nom="nombre-métrica" val="valor-métrica">` y las llamadas `<DEP id="nodo-destino" pr="pr" nl="nl">`. Pr y Nl son opcionales. Si Nl se omite, se usa Nl=Pr. Si Pr se omite, se usa Pr=1.

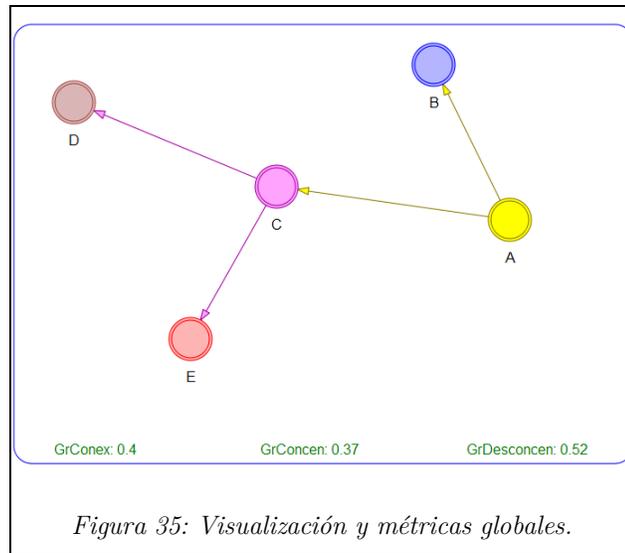


Figura 35: Visualización y métricas globales.

Como se puede apreciar, la red es más desconcentrada que concentrada. Los valores de GrConcen y GrDesconcen están afectados al reducido tamaño de la red.

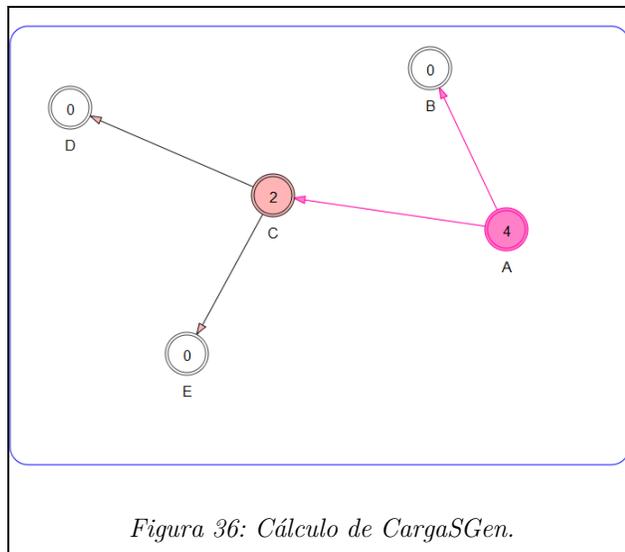
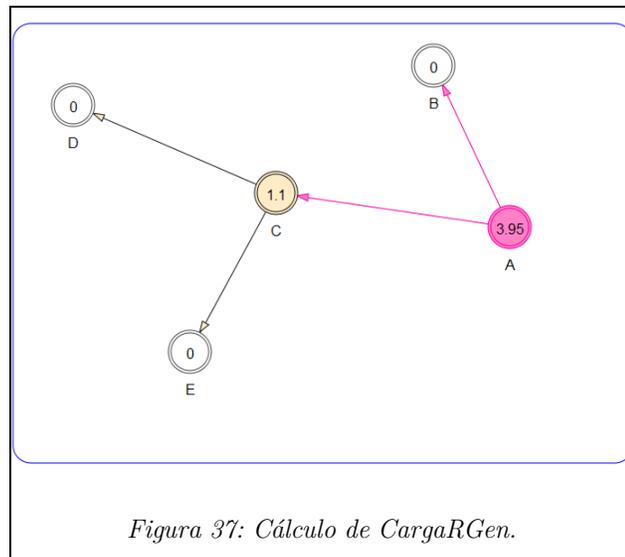
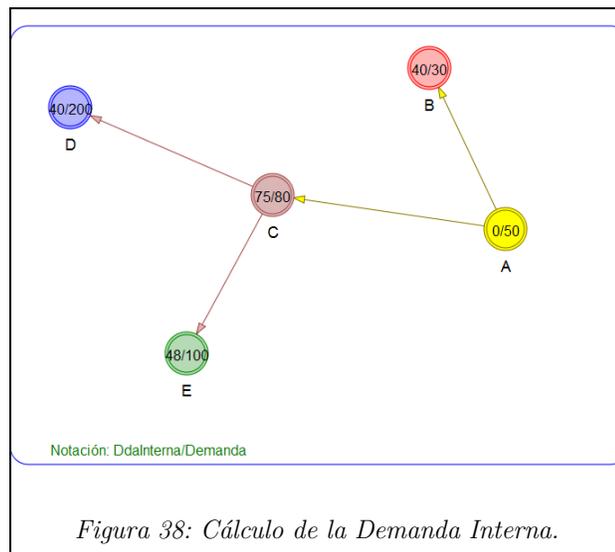


Figura 36: Cálculo de CargaSGen.

La carga estática generada muestra que, al tomar en cuenta sólo la forma del grafo, A genera toda la carga de la red. En un sistema más disperso, ello no debería ocurrir.



La carga real generada, en cambio, es más confiable que CargaSGen debido a que se dispone de más información. Se nota que, comparativamente, A sí genera mucha carga a la red. La demanda interna generada por A es 3,95 accesos.



El cálculo de la demanda interna permite apreciar cómodamente cuanta carga se genera al interior de la red. Casi toda la carga de D y E proviene del exterior. Toda la carga de A proviene

del exterior. De todas maneras, es notorio que la carga sobre C dependa casi completamente de las llamadas de A.

Como se puede notar de CargaRGen y DdaInterna, un arreglo en las dependencias de A podría hacer a esta red mucho más esbelta en términos de carga interna.

7.2 Caso práctico: Aduanas⁵⁴

7.2.1 La modernización de Aduanas

El Servicio Nacional de Aduanas⁵⁵ es un organismo del Estado con administración autónoma, relacionada al poder ejecutivo, dependiente del Ministerio de Hacienda. Entre los roles de Aduanas, están los de controlar la importación y exportación de mercancías, fiscalizando, aplicando los gravámenes pertinentes, combatiendo la evasión tributaria, el contrabando de mercancías, el tráfico de drogas, etc. sometiéndose a los requerimientos de los tratados de libre comercio. Aduanas tiene sede en Valparaíso y posee nueve sedes regionales, con las que controla alrededor de noventa puntos a lo largo del país.

La primera parte del proceso de modernización de Aduanas ocurrió entre los años 1995 y 2000. En esos años se dotó de redes computacionales al servicio, se aprobó la ley de modernización del Servicio Nacional de Aduanas (permitiendo el ingreso de datos por vía electrónica), se simplificaron los trámites aduaneros, etc. A diciembre de 1999, el 95% de las declaraciones de importación se hacía por la vía electrónica (2800 declaraciones diarias), extendiéndose el horario de 8:30 a 19:30 con permiso de retirar mercancías en una hora (antes era de 8:30 a 9:30 horas, con respuestas al día siguiente), etc. Desde esta reforma, la modernización de Aduanas se ha vuelto un proceso sostenido, destacando por sus desarrollos informáticos [Rivera 2003].

7.2.2 Redes de servicios web en Aduanas

Aduanas posee bastantes servicios web, logrando una buena interoperabilidad con varios organismos. Esto se puede ver con el proyecto “Ventanilla Única de Comercio Exterior”, que consiste en la integración de 18 organismos a Aduanas.

Un esquema general se muestra en la red de grupos presentada en la figura (basada en información obtenida de Aduanas y [PRYME 2006]).

54 Se agradece la colaboración de Cristián Vivanco de Aduanas.

55 Para más información, ver <http://www.aduana.cl> y [Rivera 2003].

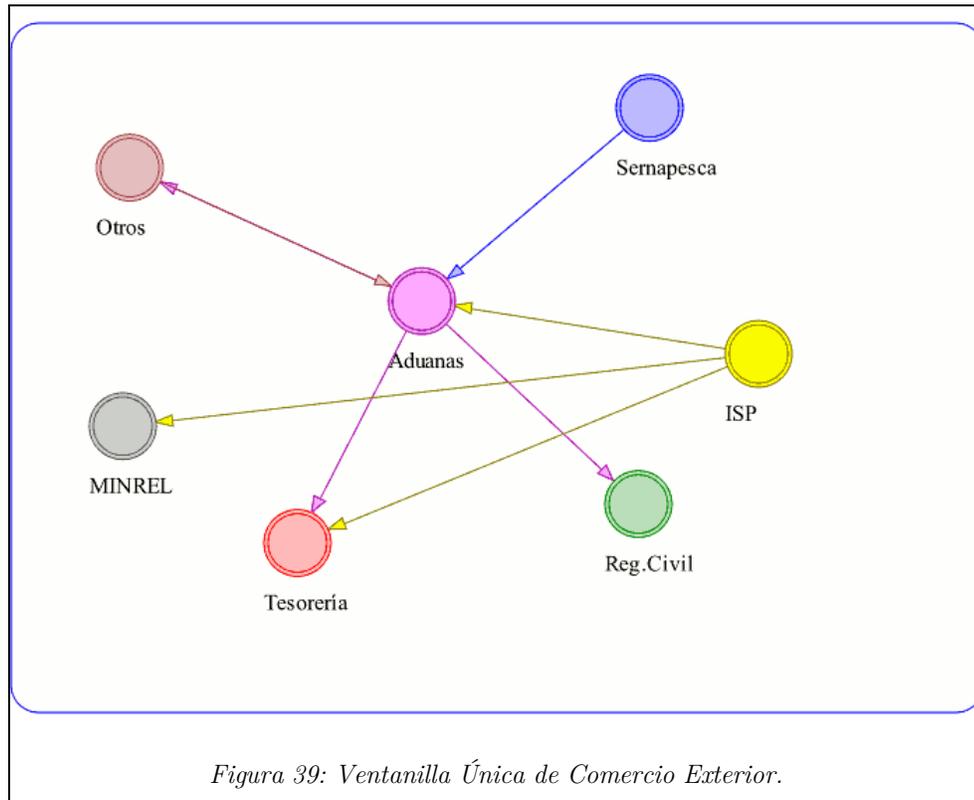


Figura 39: Ventanilla Única de Comercio Exterior.

El estado actual de la integración está muy por debajo de las 18 instituciones. En lo que sigue, se presentarán algunas de las redes de servicios web de Aduanas y las experiencias relacionadas.

La interconexión con Sernapesca ha estado sometida a restricciones. Uno de los servicios de Sernapesca accedía al servicio web de Aduanas, dejando información. Inicialmente, se disponía la información en un FTP⁵⁶. Luego la mejora fue la utilización de un servicio web que se conectaba a una base de datos (figura 40).

⁵⁶ Protocolo de Transferencia de Archivos.

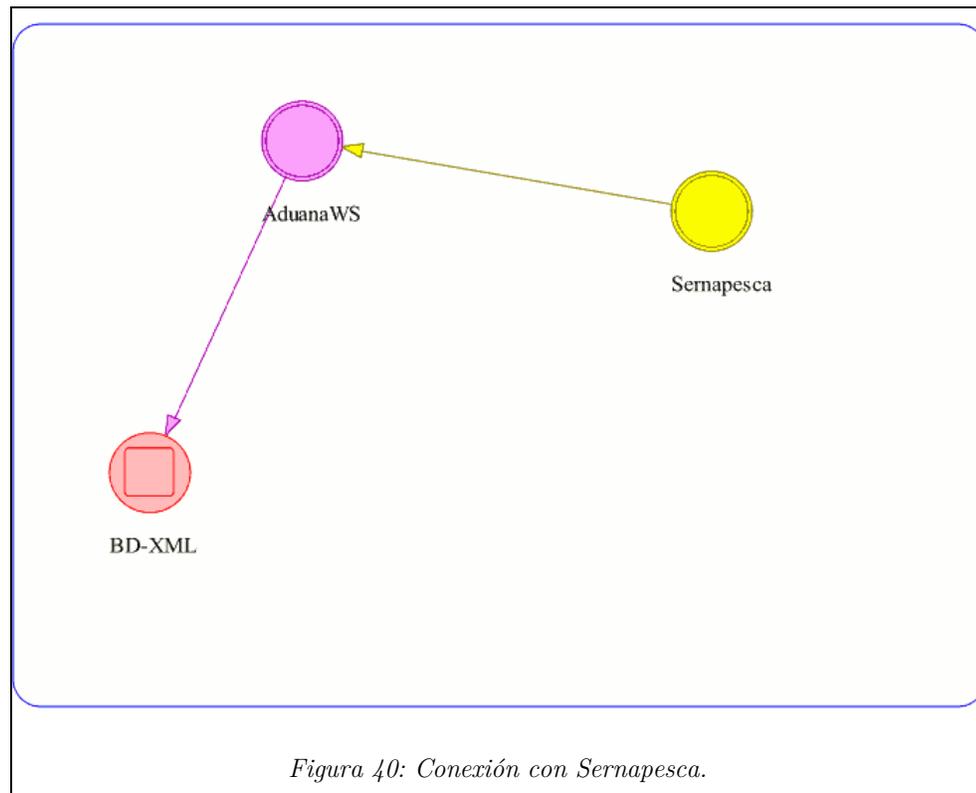
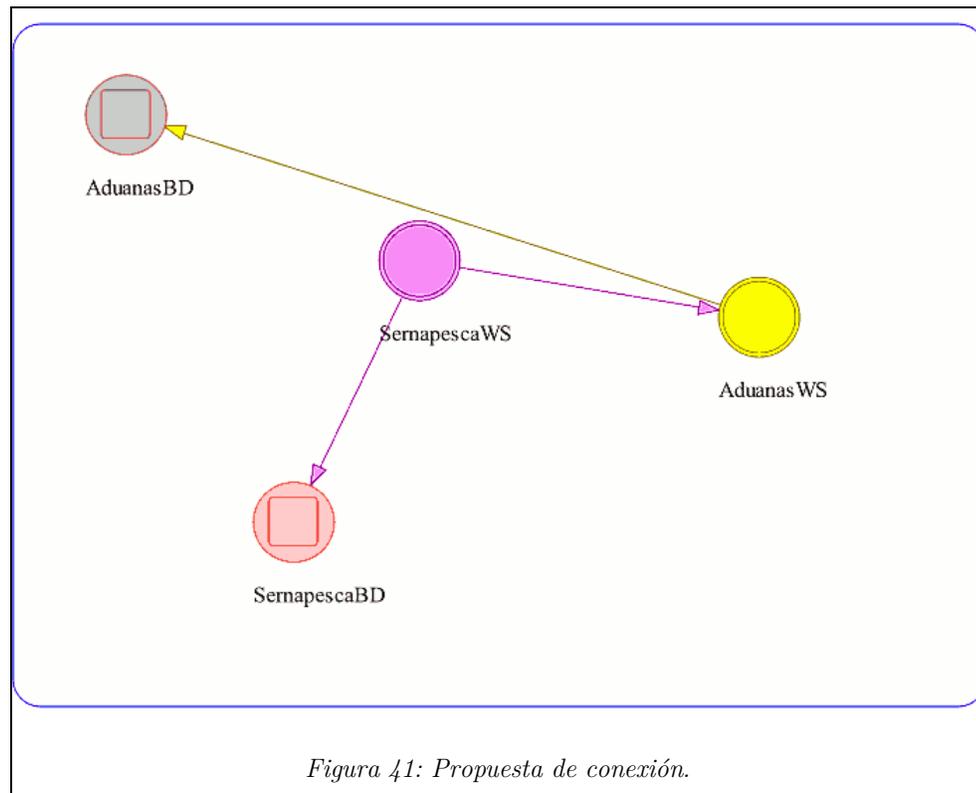


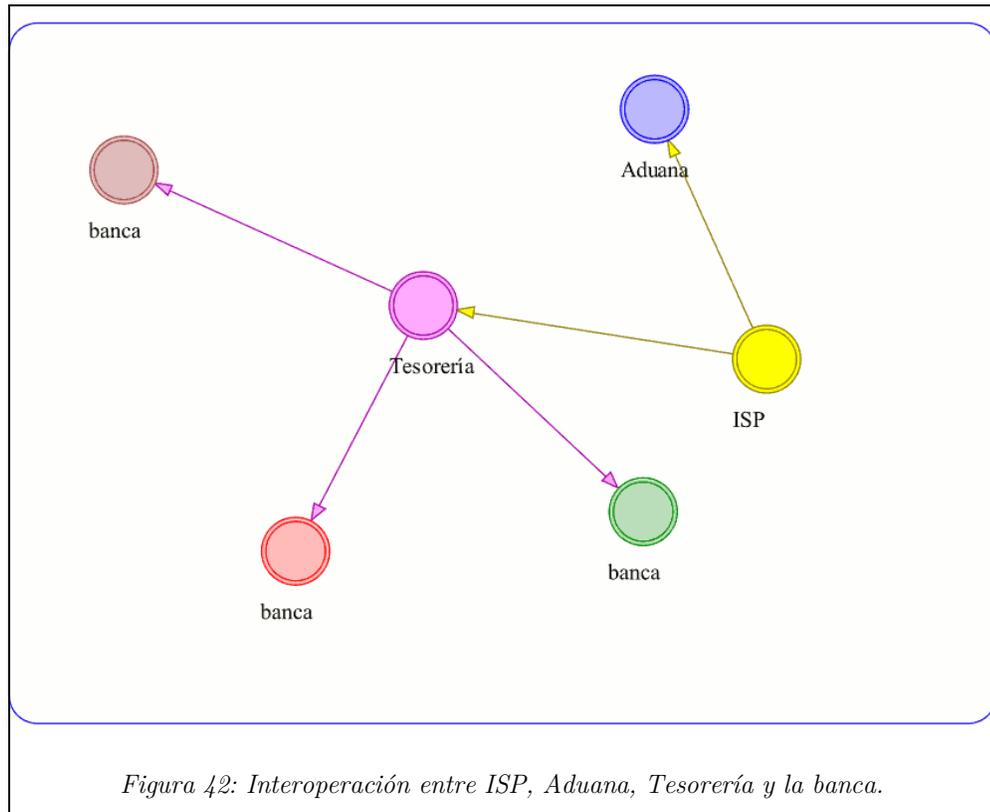
Figura 40: Conexión con Sernapesca.

En un intento por mejorar las funcionalidades, se propuso otra estructura alternativa. La figura 41 describe la propuesta de interoperación entre Sernapesca y Aduanas. Sin embargo, apareció el problema de la carga generada. Sernapesca recibe la carga de Aduanas en sus sistemas informáticos. No menor es el hecho que Aduanas tiene una demanda en sus sistemas que supera las mil transacciones diarias en varios tipos de servicios; miles de vehículos atraviesan las fronteras, miles de cargamentos entran y salen del país en un día. En esta nueva arquitectura, Sernapesca tiene más consecuencias en rendimiento debido a la carga generada. Y al carecer de una red informática avanzada, no puede asumir ese costo. Por lo tanto, la arquitectura propuesta no fue llevada a la práctica.



La arquitectura de la figura 41 no varía sustancialmente de la figura 40. La carga generada es ligeramente mayor (para obviar la demanda, vemos que CargaSGen es de 3 contra 2). El problema es un área de informática de tamaño restringido. Este problema se repite en otros organismos, lo que dificulta enormemente el progreso de la Ventanilla de Comercio Exterior (y la interconexión de Aduanas con otros organismos, en general). De hecho, muchos organismos ni siquiera tienen áreas de desarrollo informático o áreas de informática, una importante barrera para la interoperabilidad de las instituciones.

En cuanto a una red que muestra buena interoperabilidad institucional, que incluso conecta públicos y privados, es el sistema de pagos en Tesorería. En la figura 42 se presenta una operación conjunta entre el Instituto de Salud Pública (ISP), el Servicio Nacional de Aduanas, Tesorería General de la República y la banca. Tesorería está conectada a la banca debido a la necesidad de realizar pagos de cheques.



Aunque se presentó una red de grupos, esquemática, se puede acotar el propósito de la red de la figura 42 y “bajar” al nivel de servicios web (de grupos pequeños, de dos o tres servicios nada más). En el servicio web relacionado a pagar, Tesorería debe llamar a la banca con probabilidad 1 (siempre), luego las métricas Pr de las llamadas deben sumar 1. La carga al interior de cada banco es desconocida, y no interesa (está fuera del alcance de información). La carga realizada en el servicio web de consulta de Aduanas debe ser igual que el resto. Y en el trámite específico, el servicio web del ISP debe llamar al servicio web de Aduanas y al servicio web de Tesorería obligatoriamente (por necesidades de información y operación), por lo que cada llamada ocurre con probabilidad 1 ($Pr=1$).

7.2.3 Otras experiencias

De los potenciales problemas de una red de dependencias de servicios web indicados en esta memoria, Aduanas ha tenido experiencia en varios. Aquí se indican:

- *Una implementación subóptima de uno o más servicios web:* Este problema fue sufrido cuando recién se comenzaron a implementar los servicios web.

- *Un algoritmo ineficiente puede restar rendimiento a un servicio:* Si bien es claro que afecta, no hay mayor experiencia al respecto.
- *Malas prácticas de programación podrían incidir en un uso excesivo de tiempo y otros recursos de un sistema:* La experiencia aquí es **no enviar información de sobra en un servicio web**. Si distintos requerimientos de información son necesarios, que se especifique en un mensaje al servicio web. No se gana nada al enviar grandes documentos XML que sobrecargan la red y que provocan demoras en el receptor, debido a su interpretación.
- *La sobrecarga del servidor, a causa de una gran demanda de sus servicios web, podría reducir el rendimiento de cada servicio:* Aduanas posee servidores con mucha carga de trabajo debido a la excesiva demanda.
- *Hay servicios que son demasiado demandados:* Aduanas “vive” con esto. Hay servicios cuyas demandas pueden alcanzar las decenas de miles en un día.
- *Dependencias ineficientes; un servicio web podría estar siendo solicitado de manera redundante:* La experiencia aquí es que **sólo aquel organismo que posea la información, que preste el servicio web**. Las dependencias ineficientes, en las que la carga de las consultas se hacía a través de otros servicios (de manera poco transparente), era motivo de redundancia y un costo excesivo en el tiempo de respuesta de los servicios. El acceso directo a la información evita esta redundancia.
- *Hay servicios que son cruciales; si su rendimiento decae, gran parte de la red también lo hará:* Por ejemplo, esto ocurre cuando se realizan actualizaciones en los sistemas del Registro Civil e Identificación. Ello ocurre el fin de semana, justo en el momento que más vehículos atraviesan la frontera. Frente a la imposibilidad de identificar vehículos chilenos entrantes, se vuelve al método anterior (mucho más lento, burocrático), lo que genera mucho malestar entre los automovilistas.
- *Vaivenes de la conexión:* Aduanas ha tenido problemas con su conexión en el sur del país. La conexión sufre de *latencia*: la conexión a veces es rápida, a veces es lenta, a veces no se realiza. Esto ha provocado más de una dificultad.
- *Un servicio web podría tornarse lento a causa de algún servicio del cual dependa:* Esto ha pasado en Aduanas, y se explicó más atrás.
- *Un servidor tiene una potencia insuficiente; está obsoleto:* Este es un problema

típico. Algunos organismos lo sufren: los nuevos desarrollos significan una mayor sobrecarga en los sistemas debido a la demanda creciente. Ello genera la necesidad de mejorar el hardware y de aumentar la cantidad de equipos.

- *Un servicio web se ha construido con una tecnología obsoleta:* En los organismos públicos abunda la variedad de tecnologías. En Aduanas conviven muchos tipos de soluciones diferentes, y algunas se van quedando obsoletas. En otros organismos ocurren cosas similares y otros simplemente se van quedando con soluciones de software obsoletas. La recomendación es **rotar periódicamente las tecnologías ocupadas**. Esto sirve para evitar la obsolescencia e incluso para mantenerse a la vanguardia.

7.3 Comparación con otros desarrollos teóricos

7.3.1 “Modeling Quality of Service for Workflows and Web Service Processes”

Resumen: *“Workflow management systems (WfMSs) have been used to support various types of business processes for more than a decade now. In workflows for e-commerce and Web service applications, suppliers and customers define a binding agreement or contract between the two parties, specifying Quality of Service (QoS) items such as products or services to be delivered, deadlines, quality of products, and cost of services. The management of QoS metrics directly impacts the success of organizations participating in e-commerce. Therefore, when services or products are created or managed using workflows, the underlying workflow system must accept the specifications and be able to estimate, monitor, and control the QoS rendered to customers. In this paper, we present a predictive QoS model that makes it possible to compute the quality of service for workflows automatically based on atomic task QoS attributes. To this end, we present a model that specifies QoS and describe an algorithm and a simulation system in order to compute, analyze and monitor workflow QoS metrics.”*

Este es el reporte técnico UGACS -TR-02-002, LSDIS Lab, Computer Science Department, de la Universidad de Georgia, 2002. El reporte ([Cardoso 2002]) presenta un modelo predictivo de la calidad de servicio de un flujo de trabajo (workflow) a partir de atributos de elementos atómicos del flujo (actividades). En particular, se indica que este modelo predictivo se puede aplicar a servicios web.

El énfasis de este reporte está orientado completamente a los flujos de trabajo. Sin embargo, tiene algunos puntos coincidentes con el trabajo presentado en esta memoria:

1. Los flujos de trabajos son descritos como grafos (la figura 43 es una figura presente

en ese trabajo).

2. Para absorber la dificultad de transiciones condicionales (para pasos alternativos u otras situaciones), se usan probabilidades en las transiciones. Esto es similar a las probabilidades de las dependencias (Pr) en el modelo presentado en esta memoria.

3. También se permite agrupamiento de nodos, aunque mas bien es parte de un proceso de normalización de la red. En la figura 43 se muestra la eliminación de un ciclo de un paso.

4. Las métricas están fuertemente enfocadas en el tiempo promedio de respuesta y el tiempo mínimo de respuesta, y el cociente de ambos (la eficiencia). En el trabajo presentado en esta memoria, no se trabaja con el tiempo mínimo de respuesta. Y la eficiencia se podría ver a través de la variabilidad del tiempo de respuesta. De todas maneras, en el caso de servicios web, como la operación la realiza una máquina, no hay eficiencia en tiempo de respuesta. La eficiencia recae en el código y en las dependencias.

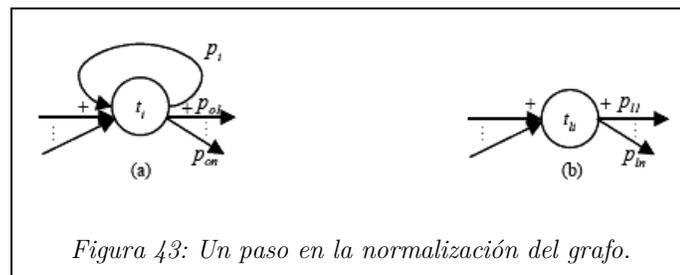


Figura 43: Un paso en la normalización del grafo.

7.3.2 “An Architecture for QoS Enabled Dynamic Web Service Deployment”

Resumen: “A system architecture for the dynamic deployment and running of Web services is presented. Service code is held in a code store; service deployment is triggered on demand by incoming SOAP messages. A mechanism is also provided for storing and applying service specific QoS targets. Policies for dynamic server allocation are developed. The system has been implemented and the results of several experiments are described.”

Este artículo ([Kubicek 2005]) fue presentado en el “UK e-Science All Hands Meeting”, AHM, 2005. Este trabajo presenta una arquitectura para trabajar con servicios web, de manera de vigilar y controlar la calidad de servicio.

La arquitectura presentada consiste en un cola para distribuir carga. Cada mensaje SOAP es un trabajo en la cola de un servicio web. Luego, el trabajo es enviado a uno de los servidores

(hay un *pool* de servidores), distribuyendo la carga. Esta arquitectura posee un *pool manager* (quien distribuye la carga) y un *pool* de servidores con servicios web.

A modo de observación, un servicio web alojado en servidores alternativos con un gestor de carga se puede modelar -usando la metodología de esta memoria- como un nodo que llama equiprobablemente a otros servicios web. La demanda interna de este modelo es balanceada.

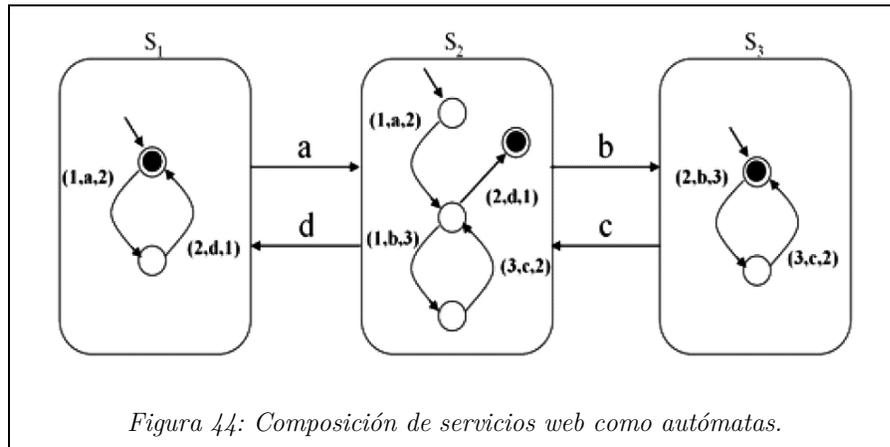
Las métricas utilizadas para monitorear la calidad de servicio son: el tiempo promedio de servicio, el porcentaje de mensajes no servido dentro de las exigencias de calidad del servicio y una métrica específica de la heurística para el balance de carga. Las dos primeras métricas están contempladas en el trabajo de esta memoria, como *TRespuesta* y *1-NServicio* (*NServicio* es la proporción de tiempos de respuesta dentro de los niveles de calidad exigidos, el complemento de la métrica usada en el artículo discutido).

7.3.3 “An Automata based approach to e-Government cooperation”

Resumen: *“With the introduction of Information and Communication Technology and with the spread of applicative cooperation between Public Administrations, it is attending an innovation process that provides to citizens and firms new electronic services of high quality and permits to Government to digitalize its procedures taking advantage of resulting scale economies. This paper faces the problem of service composition and propose a formal conceptualization to facilitate the realization, the management and the verification of applicative cooperation in the contest of e-Government. In particular, the formalization of services is achieved by fixing a finite-state automata network that permits to study the global behaviour of services and to realize possible composition taking advantage of service attributes to allow formal verification.”*

Este artículo ([Riganelli 2006]) corresponde a una presentación en la conferencia eGovInterop 2006, Bordeaux, Francia. El artículo presenta un marco de trabajo para la composición de servicios web (las dependencias de esta memoria), desde una perspectiva algorítmica.

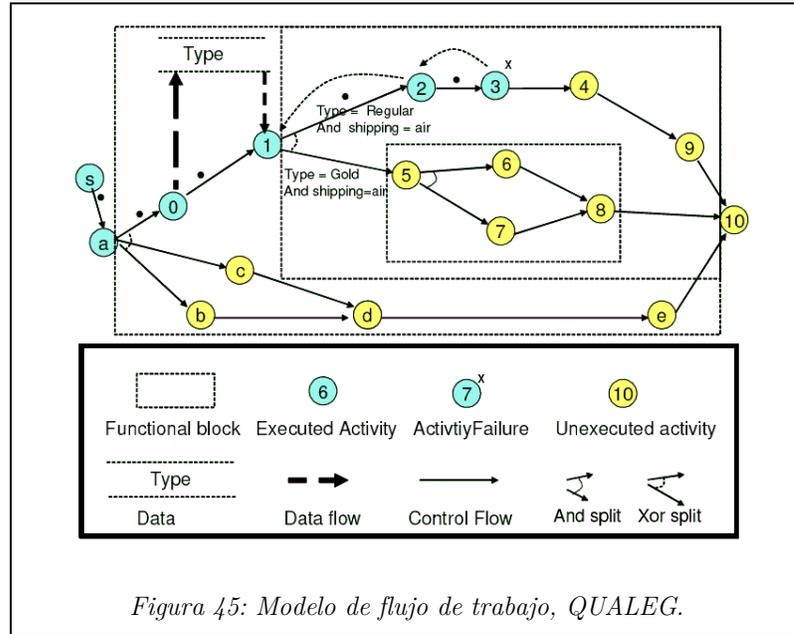
Básicamente, este trabajo modela a cada servicio web como una máquina de estados finito, un autómata no determinista (para las operaciones condicionales). En el siguiente paso, se indica que la composición de servicios web consiste en la formación de un autómata más grande, que incluye los autómatas de los servicios web (figura 44). El trabajo, sin embargo, no va más allá de eso.



7.3.4 “Research report on interoperability and co-ordination of Web Services”

Resumen: “(...) In this work we propose an approach that introduces machine-assisted design of exception handlers. We identify functional blocks in workflow business model graphs and use them in generating a minimal rollback route that is utilized in identifying an alternative path for a successful completion of a business process. We also provide an analysis mechanism that identifies the minimal changes to workflow parameters that are needed for an alternative path to be executed. The algorithm can then identify the roles that can change these parameters. Thus, the final outcome of this process is a possible exception handler model to be proposed to a human designer/operator. Such an approach can be utilized to minimize manual authoring of exception handlers. A designer can define compensating flows to regular activities in the business process (a process which has linear complexity in the number of regular activities). Then, the automated mechanism performs an analysis of the model and generates an exception handler for the modeler to evaluate. During design phase, the system can also analyze a model and can suggest exception handlers when the need arises. Our approach assists in minimizing the number of exceptions by assigning a set of activities with a single exception handler. In addition, the use of a utility model to compute the cost of exception handler alternatives provides a theoretical basis for educated design decisions. (...)”

Este es un reporte ([QUALEG 2005]) que corresponde a parte del trabajo realizado en QUALEG, *Quality of Service and Legitimacy in eGovernment*, una iniciativa europea para apoyar el desarrollo del gobierno electrónico a través de soluciones de software. QUALEG es parcialmente financiado por el programa Information Society Technology, eGovernment Unit, de la Comunidad Europea.



En este reporte se modela el flujo de trabajo como un problema de flujo en redes (un tema de grafos), en el que se buscan puntos críticos para cuando hayan fallos. De hecho, este trabajo se orienta en el control de excepciones en las actividades del flujo de trabajo.

Lo único común que tiene el trabajo discutido con el desarrollo presentado en esta memoria, es la utilización de grafos para el modelamiento y el enfoque sobre los posibles fallos de la red. De todas formas, en esta memoria el énfasis ha sido de controlar los riesgos a través de la estructura de la red, no a través de la gestión de las excepciones como en QUALEG.

7.3.5 Comentarios

La variedad de trabajos similares al modelamiento propuesto en esta memoria (la red de dependencias) valida, de alguna manera, el enfoque de grafos usado y varias métricas de rendimiento de servicios web.

Las ventajas de la metodología de modelamiento presentada en esta memoria son:

- Sólo se requiere conocer un servicio web y sus dependencias, no es necesario conocer en detalle el diagrama de flujo del servicio web (su código).
- Las métricas presentadas cubren casi todas las métricas presentadas en otros modelos. Y se presentan muchas más.

- Presenta directa relación entre las métricas, la calidad de servicio y el impacto económico de las mejoras.
- Es el más extensible de los modelos presentados.
- Esta orientado a muchos tipos de actores diferentes.

8 Comentarios y conclusiones

La interoperabilidad de la información es un requisito para un gobierno electrónico funcional, en el que las instituciones trabajan de forma cooperativa ayudándose de las tecnologías de la información y las comunicaciones. De ahí la necesidad de medir estos aspectos

Se cumplieron todos los objetivos de este trabajo:

1. Se desarrolló un modelo conceptual de análisis del gobierno electrónico. Este modelo, basado en la clasificación del tipo de trabajo en los organismos públicos, indica cuan automatizable es cada tipo de trabajo y cómo apoyarlo efectivamente con las NTIC.
2. Se desarrolló un modelo matemático, formal, para el análisis de las redes de servicios web. Basándose en algunos fenómenos económicos de las redes y las características de una arquitectura SOA, se planteó el modelamiento de las redes de servicios web como grafos dirigidos y se definió una serie de métricas para apoyar el mantenimiento y la mejora de la calidad de servicio.
3. Las métricas desarrolladas sólo miden elementos concretos, objetivos y observables del funcionamiento de los servicios web y del modelamiento realizado. Además, las métricas fueron acompañadas de una guía de estrategias genéricas para mejorar el nivel de servicio y de guías para la evaluación del impacto de las mejoras, nunca perdiéndose el objetivo de apoyar la labor de la ingeniería de software y la gestión en la administración pública.

Los modelos desarrollados son especialmente útiles como guías para que los organismos públicos logren y mantengan un buen nivel de servicio a nivel electrónico (servicios web). De esta manera, se espera apoyar el desarrollo de un buen gobierno electrónico.

Además, el modelo matemático y las métricas sirven para vigilar la interoperabilidad de la información en el gobierno electrónico, al ser una herramienta en la detección y el control de riesgos en las redes de servicios web. Y puede servir especialmente para orientar aquellos organismos públicos con menos experiencia en desarrollos informáticos.

Sin embargo, se encontró un problema importante para el uso del modelo de redes servicios web y las métricas: el escaso desarrollo informático de algunos organismos. Esto dificulta la interoperabilidad de la información automatizada entre organismos. En esos casos ni siquiera hay servicios web, por lo que el modelo de redes y las métricas apenas pueden servir para el diseño, aunque esa labor se vería complicada por la falta de experiencia. Probablemente, la aplicabilidad de este trabajo crezca en el futuro.

9 Discusión y propuestas

Una política para catalizar el desarrollo del gobierno electrónico

Uno de los problemas encontrados durante la realización de este trabajo de título fue el escaso nivel de conocimiento informático de algunos organismos. De hecho, aún hay muchos organismos públicos sin un área de informática. No se puede esperar avanzar en el gobierno electrónico con un impedimento de este tipo.

¿Cómo solucionar este predicamento? Una estrategia centralizada parece adecuada en este caso. Estos organismos carecen de recursos y conocimientos informáticos como para entrar al gobierno electrónico. Si se crea una unidad de desarrollo de software centralizada para estos organismos, se podría acelerar su modernización. Una visión para esto es considerar que el área de informática de estos organismos es externa (la unidad de desarrollo centralizada), y que su método de conexión a los sistemas ocurre vía web. Luego, la presión de adaptación de estos organismos sería menor, pues sólo debieran volverse clientes de los desarrollos del área externa. Naturalmente, una vez que hayan alcanzado un cierto tamaño institucional, debieran comenzar a desprenderse de la unidad de desarrollo centralizada, para lograr una mayor autonomía.

Países en vías de desarrollo

El desarrollo realizado puede ser bastante útil para apoyar el desarrollo del gobierno electrónico en países en vías de desarrollo. En particular, una metodología práctica propuesta sería la siguiente:

1. Considerando el modelo de análisis del gobierno electrónico (la clasificación del tipo de trabajo y la función de las NTIC en cada cuadrante), se puede hacer una inspección de cada organismo público y observar qué aspectos del organismo quedan en cada cuadrante.
2. Definir las líneas de apoyo tecnológico en base a la clasificación.
3. En particular, para el trabajo operativo de las estructuras orientadas al servicio, el desarrollo de servicios web se puede realizar con el apoyo del modelo de redes de servicios web presentado en este trabajo.

En particular, la evolución tecnológica de los servicios podría seguir la siguiente línea de desarrollo:

1. Utilización de formato XML.
2. Utilización de servicios web tradicionales.
3. Utilización de servicios web semánticos (inteligentes).

La evolución tecnológica de las tecnoestructuras podría seguir una línea diferente:

1. Utilización de formato XML.
2. Utilización de sistemas expertos semánticos (con uso inteligente de XML).
3. Utilización de sistemas expertos semánticos en línea, con utilización de conocimientos expertos de la web semántica (uso de RDF y comunicación automatizada con otros organismos a nivel internacional).

En particular, el desarrollo de un gobierno semántico es quizás muy ambicioso para los países en vías de desarrollo, pero su alcance también significaría algún grado de desarrollo.

Medición del estado de avance del gobierno electrónico

Se pueden construir muchos criterios diferentes para medir el estado de avance en el gobierno electrónico. Una solución inmediata podría ser un sistema de puntajes por cumplimiento de metas. Sin embargo, ello requiere una planificación exhaustiva de todos los hitos de la implementación del gobierno electrónico. En contraste, se propone que la medición del avance se haga mediante la evaluación del beneficio social explicado por la implementación realizada.

¿Cómo medir el estado de avance usando una evaluación socio-económica? Como son proyectos de extensión, la evaluación socio-económica se debe hacer respecto de una situación anterior. Sin embargo, se propone algo diferente, medir contra una situación ideal. De esta manera, toda la evaluación se convierte en una evaluación de costos. En otras palabras, se estaría calculando el tamaño de la brecha respecto de la situación ideal. Es necesario notar que, como ésta es una comparación entre situaciones, ninguna inversión o costo puntual (no repetible en el tiempo) debiera ser tomada en consideración. Este tipo de costos se podría reflejar en otra medida.

¿Cómo se puede modelar la situación ideal? Una situación ideal puede o no ser alcanzable. Pero una situación alcanzable sirve para realizar inversiones moderadas. En cuanto a cómo modelarla, una estrategia sería el uso del modelo de cuadrantes propuesto en esta memoria y

suponer que el trabajo es automatizado. Luego, la situación se puede hacer alcanzable relajando las exigencias, o sea, suponiendo holguras en los tiempos de atención de los servicios, aumentando los costos operacionales, etc. de la situación ideal.

10 Bibliografía

10.1 Libros, artículos e informes

[Basili 2002] V. Basili, G. Caldiera, D. Rombach. "Goal Question Metric (GQM) Approach", Encyclopedia of Software Engineering (Marciniak, J.J. ed.), online version @ Wiley Interscience, John Wiley & Sons, 2002.

[Boehm 1981] B. W. Boehm. Software Engineering Economics. Prentice Hall PTR, Upper Saddle River, New Jersey. 1981.

[Bravo 2005] C. Bravo. "Implicancias del Decreto Supremo N° 81: Una historia sobre el futuro" (en línea). Proyecto de Reforma y Modernización del Estado. Presentado en el Workshop Documento Electrónico en Chile: Experiencias y perspectivas, organizado por la Iniciativa para la Interoperabilidad de la Información (IN3), Depto. de Ciencias de la Computación, Universidad de Chile. 22 de septiembre de 2005.

http://in3.cl/workshop2005/03_CristianBravo_Implicancias_Decreto_81.pdf

[Cardoso 2002] J. Cardoso, J. Miller, A. Sheth, J. Arnold. "Modeling Quality of Service for Workflows and Web Service Processes". Reporte técnico UGACS -TR-02-002, LSDIS Lab, Computer Science Department, Universidad de Georgia. Mayo de 2002.

[Contreras 2004] Eduardo Contreras. "Evaluación social de inversiones públicas: enfoques alternativos y aplicabilidad para Latinoamérica". Serie manuales, CEPAL, Naciones Unidas. Manual n° 37. Santiago, Chile. Diciembre de 2004.

[Cornejo 2006] M. Cornejo. "Coordinación de Iniciativas de Atención al Usuario en la Administración Pública Chilena" (en línea). Documentos de Trabajo, Instituto de Asuntos Públicos (INAP), Universidad de Chile. Doc. N° 5, marzo de 2006.

<http://www.inap.uchile.cl/gobierno/destacado161.html>

[Criado 2002] J. Criado, M. Ramilo, M. Salvador. "La Necesidad de Teoría(s) sobre Gobierno Electrónico. Una propuesta integradora" (en línea). XVI Concurso de Ensayos y Monografías del CLAD sobre Reforma del Estado y Modernización de la Administración Pública. Caracas, 2002.

http://www.cnti.ve/cnti_docmgr/sharedfiles/gobiernoelectronico4.pdf

[DCC-1 2005] C. Gutiérrez, M. C. Bastarrica, R. Frez, C. Fuenzalida, S. Ochoa, P. O. Rossel, A. Villena. "Gobierno Electrónico en Chile, Desafíos, Perspectivas y Oportunidades." Revista Chilena de Ingeniería, Anales del Instituto de Ingenieros de Chile. 2005.

[DCC-2 2005] C. Gutiérrez, M. C. Bastarrica, R. Frez, C. Fuenzalida, S. Ochoa, P. O. Rossel, A. Villena. "eGovernment in Chile and the Adoption of the XML as Standard." Electronic Government - Workshop and Poster Proceedings of the Fourth International EGOV Conference 2005. Páginas 110-120. Copenhagen,

Dinamarca. Agosto de 2005.

[Doña 2006] K. Doña Molina. “Síntesis del proceso de modernización del Estado en Chile (1994-2003)” (en línea). Documentos de Apoyo Docente, Instituto de Asuntos Públicos (INAP), Universidad de Chile. Doc. N° 6, marzo de 2006.

<http://www.inap.uchile.cl/gobierno/publicaciones-dad.html>

[eGIF 2005] e-Government Interoperability Framework: Version 6.1 (eGIF) (en línea). Reino Unido. Cabinet Office. 18 de marzo de 2005.

<http://www.govtalk.gov.uk/interoperability/egif.asp>

[Erl 2004] T. Erl. “Service-Oriented Architecture”. Prentice Hall PTR, Pearson Education. 2004.

[Fenton 2000] N. Fenton, M. Neil. “Software Metrics: Roadmap” (en línea). Presentado en Proceedings of the Conference on The Future of Software Engineering, International Conference on Software Engineering. Limerick, Irlanda. 4 al 11 de junio de 2000.

<http://www.cs.ucl.ac.uk/staff/A.Finkelstein/fose/finalfenton.pdf>

<http://uweb.txstate.edu/~mg43/CS5391/Papers/Metrics/roadmap.pdf>

[Gutiérrez 2005] P. Gutiérrez. “Plataforma Integrada de Servicios Electrónicos del Estado” (en línea). Ministerio Secretaría General de la Presidencia. Presentado en el Workshop Documento Electrónico en Chile: Experiencias y perspectivas, organizado por la Iniciativa para la Interoperabilidad de la Información (IN3), Depto. de Ciencias de la Computación, Universidad de Chile. 22 de septiembre de 2005.

[HACIENDA 1999] Recursos Humanos en el Sector Público: Experiencias, análisis y propuestas. Comité Interministerial de Modernización de la Gestión Pública⁵⁷. Editado por la Dirección de Presupuestos, Ministerio de Hacienda. 1999.

[Heeks 2006] R. Heeks. Implementing and Managing eGovernment: An International Text. SAGE Publications, Athenaeum Press, Gran Bretaña. 1^a edición, 2006.

[infoDev 2002] The E-Government Handbook for Developing Countries. Center for the Democracy & Technology (CDT) y Information for the Development Program (infoDev), patrocinado por el Banco Mundial. Noviembre de 2002.

<http://www.infodev.org/content/library/detail/841/>

http://www.infodev.org/files/1064_file_eGovernment_Handbook.pdf

[Kaner 2004] C. Kaner, W. Bond. “Software Engineering Metrics: What Do They Measure and How Do We Know?” (en línea). 10th International Software Metrics Symposium (Metrics 2004). Chicago, IL. 14 al 16 de septiembre de 2004.

<http://citeseer.ist.psu.edu/701430.html>

⁵⁷ El lugar del “Comité Interministerial de Modernización de la Gestión Pública” ahora lo toma el Proyecto de Reforma y Modernización del Estado, el PRYME.

- [Kaplan 1992] R. Kaplan, D. Norton. "The balanced scorecard". Publicado en la revista Harvard Business Review. Harvard Business School Press, Boston. Enero-febrero de 1992.
- [Klischewski 2004] R. Klischewski. "Semantic Web for e-Government" (en línea). Lecture Notes in Computer Science. Springer Berlin/Heidelberg. Vol. 2739/2004, pp. 288-295. 2004.
http://is.guc.edu.eg/uploads/egov2003_klischewski.pdf
- [Kubicek 2005] C. Kubicek, M. Fisher, P. McKee, R. Smith. "An Architecture for QoS Enabled Dynamic Web Service Deployment". 4th All Hands Meeting. Reino Unido. Septiembre de 2005.
- [MIDEPLAN 2006] Precios Sociales para la Evaluación de Proyectos. Proceso 2007. Sistema Nacional de Inversiones, Ministerio de Planificación.
<http://sni.mideplan.cl/links/files/herramientas/precios-sociales/849.pdf>
- [Musso 2005] F. Musso. "I. El Gerente Público en Chile, II. Delegación y Retrodelegación de la Función Directiva" (en línea). Documentos de Apoyo Docente, Instituto de Asuntos Públicos (INAP), Universidad de Chile. Doc. N° 1, octubre de 2005.
<http://www.inap.uchile.cl/gobierno/publicaciones-dad.html>
- [Pascual 2003] P. J. Pascual. E-Government. UNDP Asia Pacific Development Program (UNDP-APDIP) y e-ASEAN⁵⁸ Task Force. Mayo de 2002.
<http://eprimers.apdip.net/series/info-economy/e-gov-toc>
- [PRYME 2003] Gobierno Electrónico en Chile: Estado del Arte. Chile. Proyecto de Reforma y Modernización del Estado, Ministerio Secretaría General de la Presidencia, Gobierno de Chile. Abril de 2003.
- [PRYME 2006] Gobierno Electrónico en Chile 2000-2006: Estado del Arte 2. Proyecto de Reforma y Modernización del Estado, Ministerio Secretaría General de la Presidencia, Gobierno de Chile. Enero de 2006.
- [QUALEG 2005] Research report on interoperability and co-ordination of Web Services. QUALEG. 2005.
- [Raad 2006] A. M. Raad. "Exclusión Digital: Nuevas Caras de Viejos Malestares". Publicado en la revista Mad (Magíster de Antropología y Desarrollo), Departamento de Antropología, Universidad de Chile. N° 14, mayo de 2006.
<http://csociales.uchile.cl/publicaciones/mad/14/raad.pdf>
- [Riganelli 2006] F. Corradoni, Ch. Ercoli, A. Polzonetti, O. Riganelli. "An Automata based approach to e-Government cooperation" (en línea). Presentado en la eGovINTEROP' 06 Conference (eGovernment Interoperability Observatory), Cité Mondiale, 18 Parvis des Chartrons, Bordeaux (France). Marzo de 2006.
<http://www.egovinterop.net/Res/9/T25%20Riganelli.pdf>

58 ASEAN es sigla de Association of SouthEast Asian Nations.

[Rivera 2003] E. Rivera Urrutia. “Nueva economía, gobierno electrónico y reforma del Estado”. FLACSO Chile y Editorial Universitaria. Santiago de Chile. 2003.

[Sahra 2001] S. Sahra, A. Ghafoor, P. A. Raymond. “Software Engineering Metrics for COTS-Based Systems”. *Computer*. Vol. 34, no. 5, pp. 44-50. Mayo de 2001.

<http://doi.ieeecomputersociety.org/10.1109/2.920611>

[Stamoulis 2001] D. Stamoulis, D. Gouscos, P. Georgiadis, D. Martakos. “Revisiting public information management for effective e-government services”. *Information Management and Computer Security (Emerald Journals)*. MCB University Press. Vol. 9, no. 4, pp. 146-153. 2001.

<http://www.emeraldinsight.com/10.1108/09685220110400327>

<http://dx.doi.org/10.1108/09685220110400327>

10.2 Sitios web

Iniciativa para la Interoperabilidad de la Información:

<http://www.in3.cl>

Proyecto Reforma y Modernización del Estado:

<http://www.modernizacion.cl>

Agenda Digital:

<http://www.agendadigital.cl>

Instructivo Presidencial Gobierno Electrónico:

<http://www.gov.cl/pres.html>

Gestión Pública:

<http://www.gestionpublica.gov.cl>

World Wide Web Consortium:

<http://www.w3.org>

UK GovTalk⁵⁹:

<http://www.govtalk.gov.uk>

The Center for The Business of Government, IBM:

<http://www.businessofgovernment.org>

Servicio de Impuestos Internos, SII Online:

<http://www.sii.cl>

Trámite Fácil:

<http://www.tramitefacil.cl>

eGovernment Interoperability Observatory:

<http://www.egovinterop.net>

Proyecto QUALEG (UE):

<http://www.qualeg.eupm.net>

eGovernment News:

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

⁵⁹ “GovTalk” propone estándares para el gobierno electrónico en el Reino Unido. Depende directamente de la “Cabinet Office”.

Gobierno Electrónico en medio oriente:	http://arab-e-gov.blogspot.com/
eGov Monitor:	http://www.egovmonitor.com/
Northern Ireland eGovernment Unit:	http://niegu.typepad.com/weblog/
infoDev:	http://www.infodev.org/
Digital Government:	http://digitalgovernment.org/
CLAD, Centro Latinoamericano de Administración para el Desarrollo:	http://www.clad.org.ve/
Federal Computer Week:	http://www.fdw.com

Apéndice A: Tecnologías disponibles (énfasis técnico)

El formato documental XML

XML significa *eXtensible Markup Language*, que se puede traducir a lenguaje extensible de etiquetas (*markups* o *tags*). XML es un formato de texto muy flexible, basado originalmente en SGML. Algunos ejemplos de archivos populares en formato XML son: XHTML para páginas WEB, SVG -*Scalable Vector Graphics*- para gráficos vectoriales, WML para páginas WAP, BPML para modelación de procesos de negocios, etc.

Para la versión 1.1 de XML (abril de 2004), el equipo de trabajo de XML utilizó los siguientes criterios de diseño:

11. XML deberá ser usado fácilmente en Internet.
12. XML deberá soportar una gran variedad de aplicaciones.
13. XML será compatible con SGML.
14. Deberá ser fácil escribir programas que procesen documentos XML.
15. El número de elementos adicionales a usar en XML se mantendrá en el mínimo absoluto (idealmente cero).
16. Los documentos XML deberán ser legibles por las personas y deberán ser razonablemente claros.
17. El diseño XML deberá ser preparado rápidamente.
18. El diseño XML deberá ser formal y conciso.
19. Los documentos XML deberán ser fáciles de crear.
20. La rigidez de las etiquetas XML será de mínima importancia.

Sintaxis de un documento XML

Un documento XML está compuesto de *entidades*. Una entidad puede tener *atributos*, texto o más entidades. El siguiente ejemplo (código 6) muestra una entidad “texto” que contiene el texto “Este es un ejemplo”:

```
<texto> Este es un ejemplo </texto>
```

Código 6: Ejemplo XML sencillo.

Las etiquetas `<texto>` y `</texto>` sirven para indicar dónde empieza y termina la entidad. La información que está fuera de estas etiquetas no pertenece a la entidad. El siguiente ejemplo muestra una entidad “párrafo” que contiene a la entidad “texto”:

```
<párrafo>
  <texto>
    Este es un ejemplo
  </texto>
</párrafo>
```

Código 7: Anidamiento en XML.

Nótese que XML sigue un esquema de anidación. No es posible cerrar la entidad “párrafo” antes de cerrar la entidad “texto”, pues “texto” es parte de la información de “párrafo”. El siguiente ejemplo muestra dos entidades anidadas dentro de otra:

```
<párrafo>
  <texto>
    Ahora hay dos entidades
  </texto>
  <texto>
    anidadas dentro de otra
  </texto>
</párrafo>
```

Código 8: Dos entidades <texto> dentro de <párrafo>.

Cuando una entidad no contiene texto ni otras entidades anidadas, se puede simplificar su notación. Las siguientes notaciones son equivalentes (1 = 2):

```
1) <texto> </texto>
2) <texto/>
```

Código 9: Formas equivalentes.

Como se mencionó, una entidad puede tener atributos. Por ejemplo, supongamos que la entidad “texto” tiene el atributo “color”. Un texto de color rojo se anotaría:

```
<texto color="rojo">
  Este es un texto de color rojo
</texto>
```

Código 10: Inclusión de un atributo.

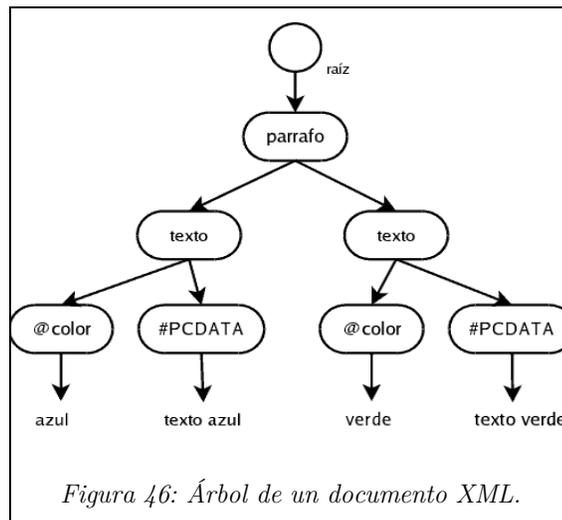
XML define relaciones entre los datos. Esto le da una importante capacidad descriptiva.

La anidación sucesiva de elementos hace que los datos en un documento XML adquieran la forma de un árbol. Obsérvese el siguiente ejemplo:

```
<parrafo>
  <texto color="azul">
    Texto azul
  </texto>
  <texto color="verde">
    Texto verde
  </texto>
</parrafo>
```

Código 11: Documento XML con aridad.

Las relaciones entre los elementos del ejemplo anterior se pueden representar en el siguiente árbol:



En el árbol anterior, “@color” identifica al atributo “color” (se usa el carácter “@” para distinguir atributos de entidades y texto). “#PCDATA”, en cambio, sólo indica la existencia de texto. No es un elemento explícito dentro de un documento XML.

Respecto a esta capacidad de representar información, dos asuntos surgen inmediatamente sobre XML:

1. La existencia de modelos de datos: no es posible relacionar datos de cualquier forma. Y en un cierto contexto, un documento XML no puede asumir cualquier estructura.
2. La manipulación de la información: un documento XML tiene relaciones entre datos, esto permite la posibilidad de *consultar* información. Y así mismo, de presentarla de otra manera, o sea, *transformarla*.

Esquema de un documento XML

Un documento XML, en un determinado contexto, no puede tener una estructura cualquiera. A esta estructura se le denomina *schema* o *esquema*. Un documento válido sigue su esquema asociado.

Existen varios lenguajes para definir el esquema de un documento XML. Entre esta lista se encuentran:

- DTD.

- RELAX.
- TREX.
- RELAX NG.
- XML Schema.
- Schematron.

Un documento XML siempre debería hacer una referencia hacia la definición de su esquema, para verificar su validez.

Además, un documento XML puede incluir varias estructuras diferentes. Debido a esto, existe la distinción de *namespaces* o “espacios de nombres”. Los espacios de nombres evitan los conflictos entre diferentes esquemas dentro de un mismo documento.

A continuación se presenta una breve selección de estas especificaciones.

DTD: DOCUMENT TYPE DEFINITION

Un documento DTD sirve para definir el esquema de un documento XML. De esta manera, sirve para verificar la validez estructural de ese documento.

Un documento DTD tiene la virtud de expresar de forma muy sencilla un esquema. Sin embargo, su poder descriptivo es bastante limitado.

Un DTD está orientado a expresar qué elementos pueden estar anidados dentro de otros y cuáles son los atributos de un elemento. En este sentido, un DTD define una *gramática* para el documento descrito, esto es, una serie de reglas de construcción para ese documento.

Un DTD está escrito en XML. Se usa el la etiqueta `<!ELEMENT>` para describir entidades y su anidación. Se usa la etiqueta `<!ATTLIST>` para definir qué atributos tiene un elemento. Por ejemplo:

```
<!ELEMENT parrafo (titulo?,texto+)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT texto (#PCDATA)>
<!ATTLIST texto color CDATA #IMPLIED>
```

Código 12: Un DTD sencillo.

En el ejemplo anterior, se indica la presencia de tres entidades: <parrafo>, <titulo> y <texto>. El significado de las líneas es el siguiente:

1. La primera línea indica que la entidad <parrafo> puede contener una entidad <titulo> (el signo “?” indica cero o uno) y debe contener una o más entidades <texto> (el signo “+” indica uno o más).
2. La segunda línea indica que la entidad <titulo> sólo puede contener texto escrito libremente (#PCDATA significa texto).
3. La tercera línea indica que la entidad <texto> sólo puede contener texto escrito libremente.
4. La cuarta línea indica que la entidad <texto> posee el atributo <color>, cuyo contenido es texto. #IMPLIED indica que la presencia de este atributo es opcional (#REQUIRED indica presencia obligada).

Como se puede ver, es muy fácil escribir un documento DTD.

RELAX NG: RELAX Y TREX

RELAX NG⁶⁰ es un lenguaje para definir el esquema de un documento XML, al igual que DTD. Relax NG es un derivado de otros dos lenguajes de definición de esquemas: RELAX y TREX, ambos basados en XML.

RELAX significa *Regular Language description for XML*. Tiene una estructura y un poder expresivo muy similar al de DTD.

TREX significa *Tree Regular Expressions for XML*. Su poder expresivo es diferente al de RELAX y DTD, pues describe los anidamientos de entidades a través de anidamientos.

60 Esta parte está basada en la información disponible en: <http://relaxng.org>

RELAX NG comprende la estructura de RELAX tanto como la de TREX (de hecho, la estructura de TREX es la dominante). A continuación se muestra un ejemplo de RELAX NG:

```
<element name="parrafo">
  <element name="titulo">
    <text/>
  </element>
  <oneOrMore>
    <element name="texto">
      <attribute name="color">
        <text/>
      </attribute>
      <text/>
    </element>
  </oneOrMore>
</element>
```

Código 13: Ejemplo sencillo de RELAX NG.

Se pueden distinguir tres entidades en este ejemplo: <parrafo>, <titulo> y <texto>. La primera entidad es <parrafo>. Un <parrafo> contiene una entidad <titulo> y una o más entidades <texto>. Una entidad <titulo> sólo contiene texto. Una entidad <texto> tiene un atributo “color” y contiene texto. El atributo “color” también consiste de texto.

XSD: XML SCHEMA

XML Schema es un poderoso lenguaje para definir el esquema de un documento XML. Es el lenguaje predominante a la hora de definir esquemas de documentos XML.

La construcción de un documento XML Schema se hace conociendo los conceptos de tipo simple y tipo complejo, que especifican la posibilidad de anidar.

Los tipos simples (simpleType) son aquellos que sólo pueden tener texto. Una entidad será simple si no tiene atributos y no permite anidar otras entidades. Un atributo siempre será simple. XML Schema permite indicar la naturaleza del texto contenido para los tipos simples, por ejemplo: string, integer, boolean, date, time, etc. o definidos según necesidad.

Los tipos complejos (complexType) permiten anidamiento. La especificación del esquema se puede hacer realizando los anidamientos o haciendo referencias a éstos.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="parrafo">
    <complexType>
      <sequence>
        <element name="titulo" type="string" />
        <element ref="texto" maxOccurs="unbounded" />
      </sequence>
      <attribute name="pagina" type="integer"
        use="optional" />
    </complexType>
  </element>
  <element name="texto">
    <complexType>
      <simpleContent>
        <extension base="text">
          <attribute name="color" type="string"
            use="required" />
        </extension>
      </simpleContent>
    </complexType>
  </element>
</schema>
```

Código 14: Ejemplo de XML Schema.

En el ejemplo anterior, la entidad `<parrafo>` contiene una entidad `<titulo>` (que es un tipo simple que sólo consta de texto) y entidades `<texto>` sin restricciones. Y tiene un atributo "pagina" que consiste solamente de un número entero, cuyo uso es opcional. Nótese que se ha usado una referencia para anidar `<texto>`; su estructura no era sencilla, así que se dejó en otra parte del documento.

La estructura de `<texto>` es más compleja. Se ha usado `<simpleContent>` para indicar que `<texto>` posee texto libremente, y que es tipo de este texto es string (a través de `<extension>`). Además, `<texto>` posee el atributo "color", de uso obligatorio, que también es de tipo texto.

Un documento XML Schema puede volverse muy complejo. Ello es natural debido a su poder descriptivo.

NAMESPACES

Un documento XML puede obedecer a varios esquemas, utilizando entidades pertenecientes a cada uno. Esto permite distinguir elementos que están representados por esquemas diferentes. Se utilizan prefijos para distinguir entre espacios de nombres. También se puede asignar un espacio de nombres por defecto (no requiere prefijo). El siguiente ejemplo incorpora dos espacios de nombres para distinguir un programa de televisión de un programa computacional:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tv:programa xmlns:tv="http://www.noexiste.org/tv.dtd">
  <tv:nombre>Vanguardia tecnológica</tv:nombre>
  <tv:duracion>44 minutos</tv:duracion>
</tv:programa>
<pc:programa xmlns:pc="http://www.noexiste.org/pc.xsd">
  <pc:nombre>Programa de grabación de audio</pc:nombre>
  <pc:tamaño>160 kilobytes</pc:tamaño>
</pc:programa>
```

Código 15: Ejemplo de namespaces.

El uso de prefijos es bastante útil: permite identificar qué trozos de un documento obedecen qué esquema y evita los conflictos de alcance de nombres.

Manipulación de documentos XML

World Wide Web Consortium ha definido una multitud de lenguajes para la manipulación de documentos XML. Actualmente existen muchas aplicaciones dedicadas a usar estas tecnologías.

Históricamente, el proyecto original para la manipulación de documentos XML consistió en la creación de XSL, *XML Stylesheet Language*. La aplicación de un estilo consiste en transformar la información de un documento y formatearla para su presentación. De este proyecto nació XPath -un lenguaje de navegación de contenidos-, XSLT -un lenguaje de transformación de documentos XML- y XSLFO -un lenguaje para transformar documentos XML en otros formatos, como PDF-. Usando nociones similares, o extendiéndolo, aparecen XQuery -un lenguaje de consulta-, XLink y XPointer.

A continuación se presenta una breve selección de estas especificaciones.

XPATH

XPath es un lenguaje para navegar dentro de un documento XML. Su sintaxis hace honor a su nombre: define “rutas” para acceder a la información. No está escrito en XML. XPath es usado por otros lenguajes, como XQuery y XSLT.

Sea el siguiente ejemplo:

```
<arbol nombre="naranja">
  <fruto>naranja</fruto>
</arbol>
<arbol nombre="higuera">
  <fruto>higo</fruto>
</arbol>
...
```

Código 16: Árboles y frutos.

Una consulta XPath para obtener todos los frutos sería: `/arbol/fruto`

La consulta anterior representa una *ruta absoluta*. Iniciar con “/” significa buscar desde la raíz. La sintaxis es muy similar a la de las rutas de archivos.

Una consulta XPath para obtener el árbol cuyo nombre es “higuera”, sería: `/arbol[@nombre="higuera"]`

La consulta anterior utiliza una condición sobre el atributo nombre de los nodos arbol.

Una consulta XPath para obtener el nombre del árbol cuyo fruto es la naranja, sería: `/arbol[fruto="naranja"]/@nombre`

XPath contiene una gran cantidad de funciones para la selección de información de sus nodos. Esto da a XPath un gran poder de navegación, lo que se complementa muy bien a su sintaxis sencilla.

XQUERY

XQuery es un lenguaje de consulta de documentos XML. Extiende XPath, usando mucha de su notación, y permite estructuras de consulta mucho más elaboradas, similares a las consultas SQL, *Standard Query Language*.

Siguiendo el ejemplo de la sección anterior, y suponiendo que el documento XML está contenido en arbol.xml, la consulta XPath `/arbol/fruto` se transformaría a XQuery de forma literal a `doc("arbol.xml")/arbol/fruto`. Toda consulta XPath se puede convertir a XQuery así.

Sin embargo, XQuery entrega la posibilidad de realizar consultas complejas a través de la sintaxis “FLOWR”, muy similar a la de SQL. La forma de una consulta FLOWR es:

```
for nodos seleccionados a través de XPath
let definición
where condición sobre los nodos
order by condición de ordenamiento
return formato de respuesta
```

Código 17: FLOWR (XQuery).

Por ejemplo, la consulta XPath `/arbol[fruto="naranja"]/@nombre` podría escribirse como:

```
for $x in doc("arbol.xml")/arbol
where $x/fruto="naranja"
return $x/@nombre
```

Código 18: Consulta FLOWR.

En la consulta anterior, `return` se puede usar para generar transformaciones. Por ejemplo, para generar una lista HTML con los frutos:

```
<ul> {
for $x in doc("arbol.xml")/arbol
where $x/fruto="naranja"
return <li> data($x/fruto) </li>
} </ul>
```

Código 19: Consulta FLOWR con formato HTML.

El uso de paréntesis de llave “{“ y “}” permite distinguir entre diferentes partes de una consulta XQuery. También permite anidarlas.

Como se ha podido notar en este par de ejemplos, XQuery permite consultar y formatear la respuesta de las consultas, entregando una gran utilidad a la hora de manipular documentos XML.

XSLT

XSLT significa *XSL Transformations*, donde XSL significa *eXtensible Stylesheet Language*. XSLT es parte de XSL. Actualmente, XSLT es usado de forma intensa en sitios web, transformando documentos XML a HTML adecuados al usuario (según su navegador, idioma, etc.).

Un documento XSLT está escrito en XML. Se utiliza una aplicación especial para procesarlo.

Sea el documento arbol.xml:

```
<arbol nombre="naranja">
  <fruto>naranja</fruto>
</arbol>
<arbol nombre="higuera">
  <fruto>higo</fruto>
</arbol>
...
```

Código 20: Árboles y frutos (repetido).

Para construir un documento XHTML que liste los árboles y sus frutos, se puede usar el siguiente documento XSLT (arbol.xsl):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

<xsl:template match="/">
  <xhtml>
    <body>
      <h1>Árboles y sus frutos</h1>
      <ul>
        <xsl:for-each select="arbol">
          <li> Árbol: <xsl:value-of select="@arbol"/>
            <br/> Fruto: <xsl:value-of select="fruto" />
          </li>
        </xsl:for-each>
      </ul>
    </body>
  </xhtml>
</xsl:template>

</xsl:stylesheet>
```

Código 21: Transformación XSLT.

Como se puede observar, ocurre una primera selección sobre la raíz del documento, a través de `<xsl:template match="/">`. La raíz se ha seleccionado mediante una consulta XPath. Luego, se escribe el documento de salida de forma casi literal, salvo por los elementos del espacio de nombres xsl.

La instrucción `<xsl:for-each select="arbol">` repite su contenido para cada nodo `<arbol>` encontrado en el documento a transformar. Esta consulta también se ha realizado según XPath.

La instrucción `<xsl:value-of select="fruto" />` devuelve el valor del elemento `<fruto>` contenido en el nodo `<arbol>` actual.

La notación de XSLT es bastante conveniente para realizar transformaciones. Su similitud a la gestión de flujos de otros lenguajes de programación -como C y Java- hace muy natural su escritura.

Otros elementos de control de flujo son la instrucción `<if test="expresión">` para condiciones simples y `<choose>` para selección por casos. También es posible ordenar una instrucción `<for-each>` a través de `<sort select="expresión" />`. Además, es posible definir variables con `<variable>` y parámetros con `<param>`.

Los servicios web

Desde sus primeros días, la tecnologías web dieron espacio a la computación distribuida (documentos HTML invocando scripts CGI). Esta interacción fue creciendo a medida que las tecnologías disponibles fueron mejorando. En el caso específico de las tecnologías XML, estas aplicaciones de computación distribuida son los “servicios web”.

Un servicio web es una funcionalidad disponible en la web. Tanto sus entradas como sus salidas pasan por Internet. Por eso mismo, es posible que un servicio web sea cliente de otro. Y esto se facilita con la utilización de XML. En este sentido, los servicios web gozan de gran potencial para interoperar.

Arquitectura orientada al servicio: SOA

Es posible construir grandes redes de servicios web con un objetivo específico. En estas redes, los servicios web interactúan entre sí.

Una arquitectura orientada al servicio es una arquitectura de computación distribuida, en la que sus átomos constituyentes son servicios. Estos servicios operan entre sí, siendo clientes y servidores unos de otros. Así, una aplicación SOA realiza su función al utilizar cada componente.

Los desafíos de una SOA, como los de cualquier arquitectura computacional distribuida, son:

1. Los problemas salidos de la latencia y baja disponibilidad del sistema de transporte de mensajes (la comunicación).
2. La ausencia de un espacio de memoria compartido entre cliente y servidor.
3. Los numerosos problemas introducidos por escenarios de falla parcial.
4. Los desafíos de acceso concurrente a datos remotos.
5. La fragilidad del sistema si se realizan actualizaciones incompatibles a algún participante.

En general, SOA y los servicios web deben utilizarse cuando:

1. Deben operar sobre Internet cuando la disponibilidad y la velocidad no están

garantizadas.

2. No hay, o es difícil, mantener un desarrollo organizado de forma centralizada (por ejemplo, cuando es muy difícil realizar actualizaciones simultáneas).
3. Cuando los componentes deben funcionar sobre plataformas y sistemas diferentes.
4. Cuando una aplicación existente necesita ser presentada en una red, y puede ser portada a Internet.

Los estándares de la W3C para los servicios web

Existe una serie de tecnologías asociadas a los servicios web que ya forman parte de la ecología tecnológica de la web.

XML

XML resuelve bastantes retos de interoperabilidad de servicios web. Es por esto que se ha vuelto una tecnología indispensable para éstos.

SOAP

SOAP provee un marco de trabajo estándar, extensible y componible, para empaquetar e intercambiar mensajes XML, independiente del protocolo de transferencia (SMTP, HTTP, FTP, etc.).

Actualmente SOAP tiene dos significados:

1. Simple Object Access Protocol: Un mensaje SOAP representa una llamada a un método remoto (SOAP RPC).
2. Service Oriented Access Protocol: Un mensaje SOAP representa la necesidad de la interacción entre distintos servicios web.

El protocolo SOAP es bastante estándar. La mayoría de los lenguajes de programación tiene soporte para SOAP, lo que facilita mucho su implementación (y reduce la necesidad de su completo entendimiento).

WSDL

WSDL significa *Web Services Description Language*. Este lenguaje detalla el funcionamiento

del servicio web, a nivel de mensajes y funcionalidades asociadas. Está escrito en XML.

Un documento WSDL tiene cuatro partes:

- Las operaciones que realiza el servicio web. Etiqueta: <portType>. Esta parte describe las funciones que realiza el servicio y los mensajes asociados a estas funciones.
- Los mensajes usados. Etiqueta: <message>. Los mensajes son los parámetros de las operaciones.
- Los tipos de datos usados. Etiqueta: <types>. Para tener neutralidad, es conveniente definir los tipos con XML Schema.
- Los protocolos de comunicación usados. Etiqueta: <binding>. Asocia cada operación a un protocolo.

WS-CDL

WS-CDL significa *Web Services Choreography Description Language*. WS-CDL indica cómo se relacionan los servicios web para interoperar. Indica qué servicios web se comunican entre sí, entre otras cosas. WS-CDL está escrito en XML.

Un documento WS-CDL tiene la capacidad de componerse, esto es, de tomar información de otros documentos WS-CDL. Esto permite describir sistemas más grandes de interoperación de servicios web. Además, tiene soporte para estructuras de control.

WS-CDL es utilizado para reflejar procesos de negocio dentro de estructuras de servicios web.

Apéndice B: Concentración de la red

Concentración y desconcentración de la red

La noción informal de red concentración de la red, es que si algunos nodos determinan el funcionamiento de gran parte de la red, la red está concentrada en esos nodos (depende de esos nodos). En cambio, si los nodos dependen de muchos otros, la red está desconcentrada. Aquí aparece un *trade off*: una red concentrada es muy sensible a la caída de ciertos servicios web mientras que una red desconcentrada tiene servicios que generan mucha carga al resto de la red. Ahora, ambos problemas desaparecen si la red está desconectada, o sea, tiene un GrConex nulo (o pequeño). Y si GrConex es alto (cercano a uno), estos problemas se agravan.

Definición[Grado de Concentración]. Se define la métrica **GrConcen** de una red $G(W,L)$, $GrConcen: G \rightarrow [0,1)$, como:

$$GrConcen(G) = \sqrt{6 \frac{\sum_{w \in W} Entrantes^2(w)}{2Nro^3(W) - 3Nro^2(W) + Nro(W)}}$$

Definición[Grado de Desconcentración]. Se define la métrica **GrDesconcen** de la red $G(W,L)$, $GrDesconcen: G \rightarrow [0,1)$, como:

$$GrDesconcen(G) = \sqrt{6 \frac{\sum_{w \in W} Salientes^2(w)}{2Nro^3(W) - 3Nro^2(W) + Nro(W)}}$$

A continuación se explica cómo se obtienen estas fórmulas.

Obtención de las fórmulas

La suma de los salientes y la de los entrantes es la misma: el número de llamadas (arcos) en la red. Este problema es el mismo para las dos métricas.

Estás métricas son una razón entre $\sum f_w^2$ y $Máx \sum f_w^2$. La raíz es para normalizar; nótese que $\sqrt{\sum f_w^2} = \|\vec{f}\|$, si \vec{f} es el vector (f_1, f_2, \dots, f_n) .

Una primera observación consiste en que un grafo que maximiza $\sum Salientes_k^2$ no posee nodos desconectados. Para probar esto, supóngase que hay un nodo no conectado a ningún otro. Si es conectado, algún *Salientes* aumentará en una unidad. Luego, $\sum Salientes_k^2$ crecerá. Por lo tanto, ninguna configuración que maximice $\sum Salientes_k^2$ puede tener algún nodo desconectado.

Ahora, la configuración que maximiza $\sum Salientes_k^2$ tiene todos sus nodos conectados. Ahora aparece una segunda observación: la configuración maximizante de $\sum Salientes_k^2$, el grafo dirigido acíclico $G(W, L)$, permite un ordenamiento de sus nodos (w_1, w_2, \dots, w_n) tal que $\forall (w_i, w_k) \in L, i < k$. Esto es posible debido a que esta estructura, un poliárbol (grafo dirigido acíclico), permite este ordenamiento (ordenamiento topológico⁶¹).

Luego, notemos que $\sum Salientes_k^2$ es creciente sobre *Salientes*_k, para cualquier *k*. Usando el ordenamiento topológico, vemos que es posible conectar un nodo w_k con cualquier nodo $w_j, j > k$ sin generar ciclos. Entonces, maximizamos la función agregando todos los nodos posibles. Si hay *n* nodos, w_k se puede conectar a todos los posteriores, o sea, a $(n - k)$ nodos, contando a *k* desde 1. Nótese que este es el máximo *Salientes* de ese nodo. Luego, el valor máximo de *Salientes* sale de maximizar la función de esta manera.

El valor máximo de salientes es: $\sum_{k=1}^n (n - k)^2$. Reescribiendo: $\sum_{k=1}^{n-1} k^2$. Como se sabe, la suma tiene la forma: $\sum_{k=1}^m k^2 = \frac{m(m+1)(2m+1)}{6}$. Reemplazando *m* por *n - 1*, se concluye.

Luego, la fórmula para la desconcentración es:

$$GrDesconcen(G) = \sqrt{6 \frac{\sum_{w \in W} Salientes^2(w)}{2Nro^3(W) - 3Nro^2(W) + Nro(W)}}$$

La deducción es la misma para *GrConcen*.

61 http://en.wikipedia.org/wiki/Topological_sorting

Apéndice C: Estimación del valor del tiempo

Según los datos de la publicación: “Compendio Estadístico 2006: Las Estadísticas de Chile”, INE, 2006, se realizará una estimación del costo del tiempo desde la perspectiva del ciudadano promedio.

Primero, se supondrá que el valor del tiempo libre es igual al valor del tiempo de trabajo (se omitirán discusiones éticas al respecto). Considerando el ingreso per cápita de \$160.000 mensual (año 2004), entonces el valor de la hora es:

$$\frac{\$ 160000}{4 \text{ semanas} \times 5 \text{ días laborales} \times 8,5 \text{ horas diarias}} \times \Delta \% IPC_{2004}^{2006}$$

Se usarán los IPC entre junio de 2004 y 2006 (por disponibilidad de la información estadística). En 2004 era de 115,87 y en 2006 era de 123,62.

El valor de la hora es:

$$Costo_{hora} = \frac{\$ 160000}{4 \times 5 \times 8,5} \times \frac{123,62}{115,87} = \$ 1004$$

Ahora se deben sumar los costos de acceso a Internet. Se asume una penetración de Internet del 36%⁶². Asumiendo \$20.000 mensuales en banda ancha (el uso del modem es escaso, así que se despreciará aquí) y \$600/hora el precio de un cibercafé (no se disponen estadísticas de uso de infocentros o bibliotecas públicas para el acceso a Internet), se debe agregar el siguiente costo de conexión:

$$0,36 \times \frac{\$ 20000}{4 \times 5 \times 16 \times 1,4} + 0,64 \times \$ 600 = \$ 400$$

El costo del tiempo de banda ancha es el costo mensual dividido por las horas aprovechables (tiempo de actividad, sin 8 horas de sueño) y 1,4 personas por hogar en promedio.

Luego, el costo por hora del uso de Internet se estima en: \$1404 (mil cuatrocientos cuatro pesos). Se ha usado el enfoque de eficiencia por comodidad.

62 <http://www.uc.cl/noticias/ficha/pub1492.html>

Apéndice D: Códigos de la aplicación

A continuación se presentan los códigos de algunas de las páginas PHP de la aplicación.

```

red_ingresar.php
<?
session_start();
?>
<html>
  <header>
    <title>
      Red de dependencias - Ingresar red
    </title>
  </header>
  <body bgcolor='white' text='black' link='green' vlink='green' alink='red'>
    <center>
      <img src='red_logo.png'>
    </center>
    <hr color='blue'>

    <form method="POST" action="red_ingresar2.php">
    <textarea name="fuente" rows="25" cols="60">

    <?

    if ($_SESSION["nodos"]) {

      $NODOS=$_SESSION["nodos"];
      $SALIENTES=$_SESSION["salientes"];
      $METRICAS=$_SESSION["metricas"];
      $NL=$_SESSION["nl"];
      $PR=$_SESSION["pr"];

      echo "<REDDEP>\n";
      foreach($NODOS as $nodid => $nodnom) {

        echo "   <WS id=\"\$nodid\" nom=\"\$nodnom\" >\n";

        $SAL = $SALIENTES[$nodid];
        $MET = $METRICAS[$nodid];

        if ($MET) {
          foreach($MET as $met => $val) {
            if ($met == "Salientes")
              continue;
            if ($met == "Entrantes")
              continue;
            echo "     <MED nom=\"\$met\" val=\"\$val\" />\n";
          }
        }

        if ($SAL) {
          foreach($SAL as $dep) {
            $spr="";
            $snl="";
            $spr=$PR[$nodid][$dep];
            $snl=$NL[$nodid][$dep];
            if ($spr!=1) $spr="pr=\"\$spr\"";
            if ($snl!=$spr) $snl="nl=\"\$snl\"";
            echo "     <DEP id=\"\$dep\" \$spr \$snl/>\n";
          }
        }
      }
    }
  }
}

```

```

echo "    </WS>\n";
}

echo "</REDDEP>\n";

} else {

?>

<REDDEP>
  <WS id="eso2.php" nom="Posicion geografica" >
    <DEP id="eso2.php" />
    <DEP id="eso3.php" />
  </WS>
  <WS id="eso2.php" nom="Meteorologia" >
    <DEP id="eso3.php" />
    <DEP id="eso5.php" />
  </WS>
  <WS id="eso3.php" nom="Poblacion" >
    <DEP id="eso6.php" />
    <DEP id="eso4.php" />
  </WS>
  <WS id="eso4.php" nom="Poblacion" >
    <DEP id="eso6.php" />
  </WS>
  <WS id="eso5.php" nom="Poblacion" >
    <DEP id="eso6.php" />
  </WS>
  <WS id="eso6.php" nom="Poblacion" >
  </WS>
</REDDEP>

<?

}

?>

</textarea><br>
<input type="submit">
</form>
</body>
</html>

```

```

red_ingresar2.php
<?
session_start();
?>
<html>
  <header>
    <title>
      Red de dependencias - Ingresar red
    </title>
  </header>
  <body bgcolor='white' text='black' link='green' vlink='green' alink='red'>
    <center>
      <img src='red_logo.png'>
    </center>
    <hr color='blue'>

    <form method="POST" action="red_ingresar2.php">
    <textarea name="fuente" rows="25" cols="60">

```

```

<?
if ($_SESSION["nodos"]) {

    $NODOS=$_SESSION["nodos"];
    $SALIENTES=$_SESSION["salientes"];
    $METRICAS=$_SESSION["metricas"];
    $NL=$_SESSION["nl"];
    $PR=$_SESSION["pr"];

    echo "<REDDEP>\n";
    foreach($NODOS as $nodid => $nodnom) {

    echo "    <WS id=\"\$nodid\" nom=\"\$nodnom\" >\n";

    $SAL = $SALIENTES[$nodid];
    $MET = $METRICAS[$nodid];

    if ($MET) {
        foreach($MET as $met => $val) {
            if ($met == "Salientes")
                continue;
            if ($met == "Entrantes")
                continue;
            echo "        <MED nom=\"\$met\" val=\"\$val\" />\n";
        }
    }

    if ($SAL) {
        foreach($SAL as $dep) {
            $spr="";
            $snl="";
            $pr=$PR[$nodid][$dep];
            $nl=$NL[$nodid][$dep];
            if ($pr!=1) $spr="pr=\"\$pr\"";
            if ($nl!=$pr) $snl="nl=\"\$nl\"";
            echo "        <DEP id=\"\$dep\" $spr $snl/>\n";
        }
    }

    echo "    </WS>\n";
    }

    echo "</REDDEP>\n";

} else {

?>

<REDDEP>
    <WS id="eso.php" nom="Posicion geografica" >
        <DEP id="eso2.php" />
        <DEP id="eso3.php" />
    </WS>
    <WS id="eso2.php" nom="Meteorologia" >
        <DEP id="eso3.php" />
        <DEP id="eso5.php" />
    </WS>
    <WS id="eso3.php" nom="Poblacion" >
        <DEP id="eso6.php" />
        <DEP id="eso4.php" />
    </WS>
    <WS id="eso4.php" nom="Poblacion" >
        <DEP id="eso6.php" />
    </WS>
    <WS id="eso5.php" nom="Poblacion" >

```

```

        <DEP id="eso6.php" />
    </WS>
    <WS id="eso6.php" nom="Poblacion" >
    </WS>
</REDDEP>

```

```

<?
}
?>

</textarea><br>
<input type="submit">
</form>
</body>
</html>

```

red_ppal.php

```

<?
session_start();
?>
<html>
  <header>
    <title>
      Red de dependencias - Ingresar red
    </title>
  </header>
  <body bgcolor='white' text='black' link='green' vlink='green' alink='red'>
    <center>
      <img src='red_logo.png'>
    </center>
    <hr color='blue'>

    <form method="POST" action="red_ingresar2.php">
    <textarea name="fuente" rows="25" cols="60">

    <?

    if ($_SESSION["nodos"]) {

      $NODOS=$_SESSION["nodos"];
      $SALIENTES=$_SESSION["salientes"];
      $METRICAS=$_SESSION["metricas"];
      $NL=$_SESSION["nl"];
      $PR=$_SESSION["pr"];

      echo "<REDDEP>\n";
      foreach($NODOS as $nodid => $nodnom) {

        echo "    <WS id=\"\$nodid\" nom=\"\$nodnom\" >\n";

        $SAL = $SALIENTES[$nodid];
        $MET = $METRICAS[$nodid];

        if ($MET) {
          foreach($MET as $met => $val) {
            if ($met == "Salientes")
              continue;
            if ($met == "Entrantes")
              continue;
            echo "      <MED nom=\"\$met\" val=\"\$val\" />\n";
          }
        }
      }
    }
  }

```

```

}

if ($SAL) {
  foreach($SAL as $dep) {
    $spr="";
    $snl="";
    $pr=$PR[$nodid][$dep];
    $nl=$NL[$nodid][$dep];
    if ($pr!=1) $spr="pr=\"$pr\"";
    if ($nl!=$pr) $snl="nl=\"$nl\"";
    echo "      <DEP id=\"$dep\" $spr $snl/>\n";
  }
}

echo "    </WS>\n";
}

echo "</REDDEP>\n";

} else {

?>

<REDDEP>
  <WS id="eso.php" nom="Posicion geografica" >
    <DEP id="eso2.php" />
    <DEP id="eso3.php" />
  </WS>
  <WS id="eso2.php" nom="Meteorologia" >
    <DEP id="eso3.php" />
    <DEP id="eso5.php" />
  </WS>
  <WS id="eso3.php" nom="Poblacion" >
    <DEP id="eso6.php" />
    <DEP id="eso4.php" />
  </WS>
  <WS id="eso4.php" nom="Poblacion" >
    <DEP id="eso6.php" />
  </WS>
  <WS id="eso5.php" nom="Poblacion" >
    <DEP id="eso6.php" />
  </WS>
  <WS id="eso6.php" nom="Poblacion" >
  </WS>
</REDDEP>

<?
}

?>

</textarea><br>
<input type="submit">
</form>
</body>
</html>

```

red_ver.php

```

<?
session_start();
?>
<html>
<header>

```

```

<title>
  Red de dependencias - Ingresar red
</title>
</header>
<body bgcolor='white' text='black' link='green' vlink='green' alink='red'>
  <center>
    <img src='red_logo.png'>
  </center>
  <hr color='blue'>

<form method="POST" action="red_ingresar2.php">
<textarea name="fuente" rows="25" cols="60">

<?

if ($_SESSION["nodos"]) {

  $NODOS=$_SESSION["nodos"];
  $SALIENTES=$_SESSION["salientes"];
  $METRICAS=$_SESSION["metricas"];
  $NL=$_SESSION["nl"];
  $PR=$_SESSION["pr"];

  echo "<REDDEP>\n";
  foreach($NODOS as $nodid => $nodnom) {

  echo "    <WS id=\"\$nodid\" nom=\"\$nodnom\" >\n";

  $SAL = $SALIENTES[$nodid];
  $MET = $METRICAS[$nodid];

  if ($MET) {
  foreach($MET as $met => $val) {
    if ($met == "Salientes")
      continue;
    if ($met == "Entrantes")
      continue;
    echo "      <MED nom=\"\$met\" val=\"\$val\" />\n";
  }
  }

  if ($SAL) {
  foreach($SAL as $dep) {
    $spr="";
    $snl="";
    $pr=$PR[$nodid][$dep];
    $nl=$NL[$nodid][$dep];
    if ($pr!=1) $spr="pr=\"\$pr\"";
    if ($nl!=1) $snl="nl=\"\$nl\"";
    echo "      <DEP id=\"\$dep\" $spr $snl />\n";
  }
  }

  echo "    </WS>\n";
  }

  echo "</REDDEP>\n";

} else {

?>

<REDDEP>
  <WS id="eso.php" nom="Posicion geografica" >
    <DEP id="eso2.php" />
    <DEP id="eso3.php" />

```

```

</WS>
<WS id="eso2.php" nom="Meteorologia" >
  <DEP id="eso3.php" />
  <DEP id="eso5.php" />
</WS>
<WS id="eso3.php" nom="Poblacion" >
  <DEP id="eso6.php" />
  <DEP id="eso4.php" />
</WS>
<WS id="eso4.php" nom="Poblacion" >
  <DEP id="eso6.php" />
</WS>
<WS id="eso5.php" nom="Poblacion" >
  <DEP id="eso6.php" />
</WS>
<WS id="eso6.php" nom="Poblacion" >
</WS>
</REDDEP>

<?
}
?>

</textarea><br>
<input type="submit">
</form>
</body>
</html>

```

```

red_grafoB.php
<?
session_start();
?>
<html>
<header>
<title>
  Red de dependencias - Ingresar red
</title>
</header>
<body bgcolor='white' text='black' link='green' vlink='green' alink='red'>
<center>
  <img src='red_logo.png'>
</center>
<hr color='blue'>

<form method="POST" action="red_ingresar2.php">
<textarea name="fuente" rows="25" cols="60">

<?

if ($_SESSION["nodos"]) {

  $NODOS=$_SESSION["nodos"];
  $SALIENTES=$_SESSION["salientes"];
  $METRICAS=$_SESSION["metricas"];
  $NL=$_SESSION["nl"];
  $PR=$_SESSION["pr"];

  echo "<REDDEP>\n";
  foreach($NODOS as $nodid => $nodnom) {

```

```

echo "    <WS id=\"\$nodid\" nom=\"\$nodnom\" >\n";

$SAL = $SALIENTES[$nodid];
$MET = $METRICAS[$nodid];

if ($MET) {
    foreach($MET as $met => $val) {
        if ($met == "Salientes")
            continue;
        if ($met == "Entrantes")
            continue;
        echo "        <MED nom=\"\$met\" val=\"\$val\" />\n";
    }
}

if ($SAL) {
    foreach($SAL as $dep) {
        $spr="";
        $snl="";
        $pr=$PR[$nodid][$dep];
        $nl=$NL[$nodid][$dep];
        if ($pr!=1) $spr="pr=\"\$pr\"";
        if ($nl!=0) $snl="nl=\"\$nl\"";
        echo "        <DEP id=\"\$dep\" $spr $snl/>\n";
    }
}

echo "    </WS>\n";
}

echo "</REDDEP>\n";

} else {

?>

<REDDEP>
    <WS id="eso.php" nom="Posicion geografica" >
        <DEP id="eso2.php" />
        <DEP id="eso3.php" />
    </WS>
    <WS id="eso2.php" nom="Meteorologia" >
        <DEP id="eso3.php" />
        <DEP id="eso5.php" />
    </WS>
    <WS id="eso3.php" nom="Poblacion" >
        <DEP id="eso6.php" />
        <DEP id="eso4.php" />
    </WS>
    <WS id="eso4.php" nom="Poblacion" >
        <DEP id="eso6.php" />
    </WS>
    <WS id="eso5.php" nom="Poblacion" >
        <DEP id="eso6.php" />
    </WS>
    <WS id="eso6.php" nom="Poblacion" >
    </WS>
</REDDEP>

<?
}

?>

</textarea><br>

```

```

<input type="submit">
</form>
</body>
</html>

```

red_cargagen.php

```

<?
session_start();
?>
<html>
  <header>
    <title>
      Red de dependencias - Ingresar red
    </title>
  </header>
  <body bgcolor='white' text='black' link='green' vlink='green' alink='red'>
    <center>
      <img src='red_logo.png'>
    </center>
    <hr color='blue'>

    <form method="POST" action="red_ingresar2.php">
    <textarea name="fuente" rows="25" cols="60">

    <?

    if ($_SESSION["nodos"]) {

      $NODOS=$_SESSION["nodos"];
      $SALIENTES=$_SESSION["salientes"];
      $METRICAS=$_SESSION["metricas"];
      $NL=$_SESSION["nl"];
      $PR=$_SESSION["pr"];

      echo "<REDDEP>\n";
      foreach($NODOS as $nodid => $nodnom) {

        echo "   <WS id=\"\$nodid\" nom=\"\$nodnom\" >\n";

        $SAL = $SALIENTES[$nodid];
        $MET = $METRICAS[$nodid];

        if ($MET) {
          foreach($MET as $met => $val) {
            if ($met == "Salientes")
              continue;
            if ($met == "Entrantes")
              continue;
            echo "     <MED nom=\"\$met\" val=\"\$val\" />\n";
          }
        }

        if ($SAL) {
          foreach($SAL as $dep) {
            $spr="";
            $snl="";
            $pr=$PR[$nodid][$dep];
            $nl=$NL[$nodid][$dep];
            if ($pr!=1) $spr="pr=\"\$pr\"";
            if ($nl!=$pr) $snl="nl=\"\$nl\"";
            echo "     <DEP id=\"\$dep\" $spr $snl/>\n";
          }
        }
      }
    }
  }

```

```
    echo "    </WS>\n";
  }

  echo "</REDDEP>\n";
} else {
?>

<REDDEP>
  <WS id="eso.php" nom="Posicion geografica" >
    <DEP id="eso2.php" />
    <DEP id="eso3.php" />
  </WS>
  <WS id="eso2.php" nom="Meteorologia" >
    <DEP id="eso3.php" />
    <DEP id="eso5.php" />
  </WS>
  <WS id="eso3.php" nom="Poblacion" >
    <DEP id="eso6.php" />
    <DEP id="eso4.php" />
  </WS>
  <WS id="eso4.php" nom="Poblacion" >
    <DEP id="eso6.php" />
  </WS>
  <WS id="eso5.php" nom="Poblacion" >
    <DEP id="eso6.php" />
  </WS>
  <WS id="eso6.php" nom="Poblacion" >
  </WS>
</REDDEP>

<?
}
?>

</textarea><br>
<input type="submit">
</form>
</body>
</html>
```