

# COMPOSICIÓN DE SERVICIOS CONVERGENTES MEDIANTE EL USO DE PATRONES DE FLUJO DE EJECUCIÓN EN UN ENTORNO TELCO 2.0



**ANDRÉS BENAVIDES PEÑA**  
**GUSTAVO ADOLFO ENRÍQUEZ HUERTAS**

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones  
Grupo de Ingeniería Telemática  
Popayán, Noviembre de 2012



# **COMPOSICIÓN DE SERVICIOS CONVERGENTES MEDIANTE EL USO DE PATRONES DE FLUJO DE EJECUCIÓN EN UN ENTORNO TELCO 2.0**

**ANDRÉS BENAVIDES PEÑA  
GUSTAVO ADOLFO ENRÍQUEZ HUERTAS**

**Trabajo de Grado presentado a la Facultad de Ingeniería Electrónica y  
Telecomunicaciones de la Universidad del Cauca para la obtención del  
Título de**

**Ingeniero en Electrónica y Telecomunicaciones**

**Director: Juan Carlos Corrales  
Asesor: Jesús David Ramírez**

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Grupo de Ingeniería Telemática  
Popayán, Noviembre de 2012**



## **Agradecimientos**

Queremos expresar nuestros agradecimientos a nuestras familias por el apoyo incondicional durante toda nuestra vida como estudiantes universitarios. A nuestros guías durante el proceso de elaboración del Trabajo de Grado: al Ingeniero David Ramírez por su asesoría y completa disposición para sortear los obstáculos que presentó este proyecto; al Doctor Juan Carlos Corrales Muñoz por dirigir el trabajo; y en general, a todo el Grupo de Ingeniería Telemática.



# Contenido

Lista de Figuras .....	xi
Lista de Tablas .....	xiii
Lista de Siglas .....	xv
1 Introducción.....	1
1.1 Contexto General .....	1
1.2 Escenario de Motivación.....	2
1.3 Definición del Problema.....	3
1.4 Objetivos .....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos .....	5
1.5 Alcance.....	5
1.6 Contribuciones.....	6
1.7 Estructura de la Monografía.....	6
2 Estado Actual del Conocimiento.....	7
2.1 Base Conceptual .....	7
2.1.1 Entorno Telco 2.0 .....	7
2.1.2 Composición de servicios .....	8
2.1.3 Patrones de flujo de ejecución.....	9
2.1.4 Especificación JAIN SLEE .....	10
2.1.5 Plataforma de entrega de servicios.....	11
2.2 Trabajos Relacionados .....	13
2.3 Resumen .....	17
3 Clasificación de Servicios.....	19
3.1 Definiciones .....	19
3.1.1 Definiciones Telco .....	20
3.1.2 Definiciones Web.....	21
3.2 Revisión de servicios de Telecomunicaciones, Web y Convergentes .....	24
3.2.1 Organizaciones de estandarización .....	24

3.2.2	Plataformas para el despliegue de servicios .....	27
3.2.3	Operadores de telecomunicaciones .....	29
3.3	Modelo en capas de un entorno Telco 2.0.....	30
3.4	Catálogo de servicios Propuesto.....	32
3.4.1	Servicios de Telecomunicaciones .....	32
3.4.2	Servicios Web .....	34
3.4.3	Servicios Convergentes .....	36
3.4.4	Acceso a datos.....	37
3.4.5	Seguridad.....	38
3.5	Resumen.....	39
4	Detección de patrones de flujo de ejecución en servicios Telco 2.0.....	41
4.1	Análisis de patrones de Flujo de Ejecución .....	42
4.1.1	Patrones de flujo de control.....	42
4.1.2	Patrones de Datos .....	43
4.2	Modelado de servicios Telco 2.0 .....	45
4.2.1	Redes de Petri .....	45
4.2.2	Modelado de servicios en redes de Petri.....	46
4.3	Método para la detección de Patrones de Flujo de ejecución.....	51
4.3.1	Detección de Patrones de Flujo de Control .....	52
4.3.2	Detección de Patrones de Datos.....	59
4.4	Patrones de Flujo de Ejecución Presentes en servicios Telco 2.0.....	61
4.4.1	Lista de Patrones de Flujo de Control presentes en un entorno Telco 2.0.....	61
4.4.2	Lista de patrones de datos presentes en un entorno Telco 2.0 .....	62
4.4.3	Ejemplo – composición de un servicio Telco 2.0 mediante patrones de flujo de ejecución .....	63
4.5	Resumen.....	64
5	Prototipo, Experimentación y Resultados.....	67
5.1	Prototipo Funcional .....	67
5.1.1	Descripción de la herramienta GT-4SC (Graphical Tool-patterns-based for Service Creation) .....	67
5.1.2	Diagramas y Arquitectura.....	69
5.1.3	Consideraciones generales del prototipo.....	76
5.1.4	Interfaces gráficas de usuario del sistema y Descripción de Servicios Implementados .....	77
5.2	Experimentación y pruebas.....	79
5.2.1	Plan de pruebas.....	80
5.2.2	Resultados obtenidos.....	82



5.3	Resumen .....	90
6	Conclusiones, resultados y trabajo futuro .....	91
6.1	Conclusiones .....	91
6.2	Resultados .....	92
6.3	Trabajo Futuro .....	93
7	Referencias .....	95



## Lista de Figuras

Figura 1-1: Alcance del trabajo .....	5
Figura 3-1: Clasificación de servicios de telecomunicaciones; tomado de [40].....	20
Figura 3-2: Ubicación de servicios portadores y básicos [43] [44].....	21
Figura 3-3: Ciclo del modelo de creación de conocimiento en sitios Web 2.0 .....	22
Figura 3-4: Servicio Web compuesto .....	24
Figura 3-5: Capa de servicios propuesta teniendo en cuenta la arquitectura en capas de un entorno Telco 2.0 de [23] .....	31
Figura 4-1: Esquema en módulos para la detección de patrones de flujo de ejecución en servicios Telco 2.0 .....	41
Figura 4-2: Patrón de selección diferida (Deferred Choice) representado mediante redes de Petri .....	43
Figura 4-3: Modelo de Workflow para identificar DP .....	44
Figura 4-4: Patrón: copia de datos .....	45
Figura 4-5: Ejemplo de una red de Petri.....	46
Figura 4-6: Servicio de llamada básica representado mediante a) redes de Petri. ....	47
Figura 4-7: Representación genérica de un servicio Web genérico mediante a) redes de Petri b) Grafos.....	48
Figura 4-8: Representación de servicio web genérico mediante redes de Petri con posibles respuestas .....	49
Figura 4-9: Servicio Shopping representado mediante redes de Petri.....	50
Figura 4-10: Método para la detección de patrones de flujo de control .....	52
Figura 4-11: Ejemplo de grafo con nodos simples y de flujo de control .....	53
Figura 4-12: Servicio básico de llamada representado mediante redes de Petri (izquierda) y su correspondiente transformación a grafos (derecha) .....	54
Figura 4-13: Grafo TD del servicio de llamada .....	55
Figura 4-14: Proceso de detección de patrones de flujo de ejecución.....	58
Figura 4-15: Ejemplo de detección de patrones de flujo de control .....	59
Figura 4-16: Resultados de la detección de patrones .....	61
Figura 4-17: Ejemplo de detección de patrones de flujo de ejecución en el servicio "CallMessage".....	63
Figura 5-1: Módulos de la herramienta GT-4SC.....	68
Figura 5-2: Arquitectura de la herramienta.....	69
Figura 5-3: Código generado - Componentes CFP, DP .....	71
Figura 5-4: Diagrama de despliegue .....	72
Figura 5-5: Diagrama de Orquestacion .....	73
Figura 5-6: Diagrama en paquetes, subsistemas y capas del sistemas .....	76
Figura 5-7: Interfaz de Inicio .....	79
Figura 5-8: Interfaz de registro.....	79
Figura 5-9: interfaz de usuarios .....	79

Figura 5-10: Interfaz usuarios de destino .....	79
Figura 5-11: Ilustración de la Prueba T1.....	82
Figura 5-12: Ilustración de la Prueba T2.....	82
Figura 5-13: TwitterCalls Experimentación A – T1.....	84
Figura 5-14: TwitterCalls Experimentación B – T1.....	84
Figura 5-15: TwitterSMS Experimentación A – T1.....	84
Figura 5-16: TwitterSMS Experimentación B – T1.....	84
Figura 5-17: CallMessage Experimentación A – T1.....	84
Figura 5-18: CallMessage Experimentación B – T1.....	84
Figura 5-19: TwitterCalls Experimentación A y B - T2 .....	85
Figura 5-20: TwitterSMS Experimentación A y B - T2.....	85
Figura 5-21: CallMessage Experimentación A y B - T2.....	85

## Lista de Tablas

Tabla 2-1: Brechas del conocimiento .....	16
Tabla 3-1 Clasificación de servicios de la WEB 2.0, siguiendo el modelo de creación de conocimiento .....	23
Tabla 3-2: Recomendaciones de la serie F de la ITU.....	26
Tabla 3-3: Recomendaciones de la serie I de la ITU.....	26
Tabla 3-4: Recomendaciones de la serie Y de la ITU .....	26
Tabla 3-5: Catálogo de Servicios: Servicios de Telecomunicaciones - Básicos .....	32
Tabla 3-6: Catálogo de Servicios: Servicios de Telecomunicaciones - Suplementarios .....	32
Tabla 3-7: Catálogo de Servicios: Servicios de Telecomunicaciones - Compuestos.....	33
Tabla 3-8: Catálogo de Servicios: Servicios Web - Tradicionales .....	34
Tabla 3-9: Catálogo de Servicios: Servicios Web - Web 2.0 .....	35
Tabla 3-10: Catálogo de Servicios: Servicios Web -Compuestos.....	36
Tabla 3-11: Catálogo de Servicios: Servicios Convergentes.....	37
Tabla 4-1: Correspondencia entre lenguajes de modelado y el tipo de datos que soportan ..	44
Tabla 4-2: Servicios modelados en redes de Petri .....	51
Tabla 4-3: Transformación de redes de Petri a grafos .....	53
Tabla 4-4: Conjunto de patrones de objeto de estudio clasificados según su funcionalidad y comportamiento .....	56
Tabla 4-5: Patrones de datos y su correspondiente implementación con JAIN SLEE .....	60
Tabla 4-6: Patrones identificados después de la detección de patrones .....	62
Tabla 4-7: Lista de patrones de datos presentes en un entorno Telco 2.0 .....	62
Tabla 5-1: Estructura de la representación Gráfica de T1 .....	83
Tabla 5-2: Estructura de la representación Gráfica de T2 .....	83
Tabla 5-3: Análisis inferencial de los resultados obtenidos en la Prueba T1 .....	88
Tabla 5-4: Análisis inferencial de los resultados obtenidos en la Prueba T2 .....	89



## Lista de Siglas

**BP:** Business Process.

**BPEL:** Business Process Execution Language.

**BPM:** Business Process Modeling Language.

**BPML:** Business Process Modeling Language

**CAP:** CAMEL Application Part

**ECE:** Ericsson: Composition Engine

**FSA:** Finite State Automata.

**FSM:** Finite State Machine

**IMS:** IP Multimedia Subsystem

**INAP:** Intelligent Network Application Part)

**IT:** Information Technologies.

**JAIN SLEE:** Java APIs for Integrated Networks Service Logic Execution Environment

**JDBC:** Java Database Connectivity

**JMX:** Java Managment Extensions

**JPDL:** jBPM Process Definition Language

**JSON:** JavaScript Object Notation

**PLMN:** Public land mobile network

**PSL:** Property Specification Language.

**RDBMS:** Relational DataBase Management System.

**REST:** Representational State Transfer.

**SBB:** Service Building Blocks

**SDL:** Service Description Language

**SGBD:** Sistemas de gestión de bases de datos

**SOA:** Service Oriented Architecture.

**TDA:** Trace Detection Algorithm

**TIC:** Tecnologías de la Información y las Comunicaciones.

**WfMC:** Workflow Management Coalition

**WS:** Web Services

**WSCL:** Web Services Conversation Language

**WSDL:** Web Service Description Language.

**XML:** Extensible Markup Language.

**XPDL:** XML Process Definition Language.

**YAWL:** Yet Another Workflow Language.



# Capítulo 1

## 1 Introducción

### 1.1 Contexto General

Actualmente, los proveedores de servicios de telecomunicaciones (Telcos), están buscando nuevos métodos para la creación y ejecución de servicios convergentes<sup>1</sup> que permitan: reducir el tiempo de salida al mercado (Time to Market<sup>2</sup>), reaccionar de forma más rápida a las condiciones dinámicas del mercado, aumentar sus ganancias, y satisfacer las necesidades de los usuarios [1].

Uno de los métodos que soporta el proceso de creación de nuevos servicios, se denomina composición, el cual permite combinar servicios previamente desarrollados, implementados, depurados y desplegados; de tal forma que puedan ofrecer funcionalidades adicionales que un servicio, por sí solo no ofrece [2].

El proceso de composición puede ser abordado desde diferentes perspectivas; la primera se enfoca en la integración y ejecución de servicios; y la segunda busca estandarizar dicho proceso. En la primera perspectiva sobresalen las plataformas para el Despliegue de Servicios (SDP), las cuales proveen una solución integral para la creación, implementación, despliegue y monitoreo de servicios, algunos ejemplos son: el Gestor de Interacción entre Capacidades de Servicios (SCIM), el motor de Composición Ericsson (ECE), las SDP de Mobicents y Rhino, entre otras. Así mismo, la especificación JAIN SLEE [3] (Java APIs for Integrated Networks Service Logic Execution Environment) ha tenido gran acogida para el proceso de creación de servicios, ya que provee un entorno ideal para las comunicaciones y para la invocación de servicios Web a través de adaptadores de recursos.

Por otra parte, la segunda perspectiva pretende estandarizar el proceso de composición con lenguajes bien definidos, como por ejemplo BPEL (Business Process Execution Language) y JPDL (jBPM Process Definition Language); los cuales han sido utilizados principalmente en el mundo de Web, para definir como cada uno de los servicios se relaciona con otros para obtener servicios compuestos. Lo anterior se logra gracias al uso de estructuras recurrentes

---

<sup>1</sup> Servicio Convergente: Coordinación de un conjunto de servicios proporcionados mediante diferentes redes, tales que a la vista del usuario final sean ofrecidos como un solo servicio.

<sup>2</sup> Time to Market: Es un término que se refiere al tiempo que toma un servicio en ser creado, desplegado, probado y depurado, para que el usuario final lo pueda consumir.

llamadas patrones, que permiten definir el flujo de control y datos (Flujo de Ejecución)<sup>3</sup>, los cuales son necesarios para que dos o más componentes, se relacionen para cumplir una función específica.

Como se mencionó anteriormente, dichos patrones han sido utilizados para definir el flujo de ejecución de servicios compuestos en la Web, sin embargo, algunos trabajos como [4] [5] [6], proponen arquitecturas que hacen uso de los patrones soportados por los lenguajes ya mencionados, para componer servicios convergentes. Lo anterior se realiza debido a que los patrones presentan importantes ventajas para facilitar el proceso de composición; una de ellas es la reusabilidad, ya que cada uno de estos puede ser reutilizado para definir el flujo de ejecución del servicio; otra ventaja es que cada patrón permite representar un comportamiento específico [7] [8] (síncrono, asíncrono, paralelo, divergente, etc.), lo que permite capturar comportamientos propios de servicios Web y de Telecomunicaciones, para soportar la composición de servicios convergentes.

Sin embargo, a pesar de todos los adelantos que se han realizado hasta la actualidad en el campo de la composición, aún existen dificultades cuando se desean componer servicios propios del mundo de las telecomunicaciones con servicios del mundo IT<sup>4</sup>, ya que estos tienen características de desempeño, confiabilidad y disponibilidad completamente diferentes. Por esta razón, muchos grupos de investigación y empresas del sector de las comunicaciones han puesto especial interés en encontrar soluciones a estos problemas, invirtiendo capital en mecanismos que faciliten la composición de servicios, la reducción del Time to Market, y por supuesto, la satisfacción del usuario.

## 1.2 Escenario de Motivación

En los últimos años, las empresas del sector de las telecomunicaciones, han sido obligadas a reformular sus modelos de negocio tradicionales, en los cuales ofrecían un portafolio de servicios muy reducido a los usuarios, a los nuevos modelos que integran aplicaciones, servicios y contenido proveniente del lado de la Web. Esto con el fin de ofrecer un mayor número de servicios a un costo accesible, para mantener la fidelidad de los usuarios a un determinado operador.

Este nuevo modelo que incluye las capacidades de los servicios de Telecomunicaciones en combinación con servicios Web, ha sido denominado “Telco 2.0” por algunos autores como: Yoon *et al.* en [9] y algunas iniciativas como: “La iniciativa Telco 2.0” en [10]. Sin embargo, la adopción de dichos modelos, implica necesariamente una inversión significativa por parte de las empresas del sector de las Telecomunicaciones, en infraestructura y plataformas que faciliten el despliegue de los servicios.

Muchas de éstas empresas son multinacionales, las cuales poseen el capital suficiente para adquirir SDP, que cuentan con entornos de creación, ejecución, despliegue, monitoreo y facturación, de los servicios que ofrecen. Sin embargo, si nos trasladamos al contexto Colombiano, muchas pequeñas y medianas empresas (PYMES), no producen ganancias suficientes para hacer este tipo de inversiones; por ésta razón, se ven obligadas a buscar

---

<sup>3</sup> El termino flujo de ejecución hace referencia a la unión entre el flujo de control y datos.

<sup>4</sup> IT: Siglas de Tecnologías de la información (Information technology)

soluciones alternativas y reformular sus modelos de negocio para aumentar sus ingresos [11].

A pesar de los altos costos que implica la adquisición de SDP propietarias, existen plataformas de libre distribución, que las empresas con capital limitado, pueden utilizar durante el proceso de composición de los servicios que desean ofrecer al mercado [12]. Sin embargo, los servicios deben ser desarrollados a nivel de código por programadores expertos, ya que ninguna de estas plataformas cuenta con interfaces gráficas que permitan separar la lógica de negocio de la implementación, y al mismo tiempo permitan componer servicios de forma simple e intuitiva, lo que se traduce en dos problemas: aumento en el tiempo de creación y despliegue del servicio; y reducción de ganancias.

El escenario en el cual se encuentran muchas empresas prestadoras de servicios de comunicaciones, motivó el trabajo de grado, el cual propone un método para facilitar la composición de servicios utilizando lo mejor de las capacidades de las SDP de libre distribución y los mecanismos de composición, para que las empresas que tienen su fuente de ingresos en el desarrollo de servicios, encuentren una alternativa para la creación y despliegue de servicios en un Time to Market razonable.

### 1.3 Definición del Problema

En la actualidad existe una amplia gama de servicios convergentes y los requerimientos de los usuarios son cada vez más exigentes, debido a esto, los Telcos han tenido que mejorar su competitividad para crecer y mantenerse en la industria. Por las anteriores razones, estos se encuentran en la búsqueda de nuevos métodos que permitan disminuir el tiempo y los costos de creación de servicios. Una opción para alcanzar dicho objetivo, es la integración de servicios existentes para responder a necesidades complejas, lo cual se logra a través del proceso de la composición.

En este proceso se identifican claramente tres enfoques, el primero es el de la Web, con los nuevos servicios sociales (Facebook, Twitter, etc.); el segundo desde los operadores de telecomunicaciones, con servicios tradicionales (llamada, buzón de mensajes, SMS, etc.); y el tercero, es un enfoque que reúne los dos anteriores, llamado composición unificada o composición de servicios Telco 2.0 [13]. De esta forma se hace necesario definir el comportamiento de cada uno de estos servicios compuestos, el cual se hace mediante lenguajes estandarizados o mediante el uso de estructuras recurrentes llamadas patrones, que permiten definir el flujo de ejecución. A continuación se explican los tres enfoques.

- El mundo de la Web, se ha enfocado en la mecanización y automatización de métodos, tareas y secuencias de acciones [14], con el fin de obtener soluciones recurrentes que aumenten la efectividad en los procesos de composición. De esta forma se han definido un conjunto de patrones de flujo de ejecución [15] [7] [8], para los servicios compuestos en la Web, los cuales son descritos y soportados por lenguajes como BPEL, XPDL, JPDL, entre otros. La utilización de dichos patrones es una alternativa que permite la estandarización del flujo de ejecución, lo cual incrementa la eficiencia en los procesos de creación y composición de servicios [16].

- En el mundo de las telecomunicaciones, las SDP proporcionan una solución integral para la entrega de servicios a los consumidores, algunas de ellas hacen uso de la especificación JAIN SLEE [17], la cual se caracteriza por una alta disponibilidad, alto rendimiento, baja latencia, estar orientado a eventos, iteraciones asíncronas, entre otras. Además, una gran ventaja de JAIN SLEE es el estar orientado a componentes, lo que permite utilizar la composición para facilitar el proceso de creación de nuevos servicios. En este enfoque también se han definido patrones del flujo de ejecución que permiten modelar el comportamiento de un servicio mediante el uso de FSM, SDL [18], entre otros. Sin embargo, estos son escasos y no son estandarizados, además están sujetos a herramientas propietarias, lo cual es una desventaja para este tipo de entornos.
- En un entorno Telco 2.0, se han desarrollado proyectos de investigación como: TeamCOM [4], OPUCE [19] y SPICE [20], los cuales han propuesto soluciones que integran las ventajas de los dos anteriores enfoques. Sin embargo, no se ha estandarizado ninguna las soluciones que cada uno de estos propone, y además no es posible tener acceso a los entregables de dichos proyectos, debido a su carácter propietario.

Como se mencionó en cada uno de los tres enfoques, existen ventajas y desventajas, los cuales se resumen así: las plataformas basadas en JAIN SLEE permiten realizar procesos de composición para la creación de servicios, sin embargo, estos han sido desarrollados por expertos, ya que la implementación de dichos servicios se debe realizar en código fuente<sup>5</sup>. Adicionalmente, las plataformas libres JAIN SLEE no cuentan con alguna interfaz gráfica que genere el código automáticamente, lo cual se convierte en otro limitante. Por otra parte, no se han definido patrones que permitan estandarizar y soportar la composición de servicios en un entorno convergente, como los utilizados en la Web [21], los cuales permitan reducir la complejidad del proceso de composición de servicios Telco 2.0.

En este sentido, dicha composición implica un aumento en el tiempo de diseño e implementación de servicios, convirtiendo este proceso en algo: tedioso, complicado y poco eficiente.

Teniendo en cuenta el problema planteado anteriormente, surge la siguiente pregunta de investigación:

¿Cómo facilitar la composición de servicios convergentes en un entorno Telco 2.0?

## 1.4 Objetivos

### 1.4.1 Objetivo General

- Soportar la composición de servicios convergentes mediante el uso de patrones de flujo de ejecución en un ambiente Telco 2.0.

---

<sup>5</sup> El código fuente de un programa está escrito por un programador en algún lenguaje de programación.

### 1.4.2 Objetivos Específicos

- Elaborar un catálogo de servicios básicos y compuestos presentes en un ambiente Telco 2.0.
- Identificar los patrones de flujo de control y datos presentes en servicios Telco 2.0.
- Evaluar experimentalmente a través de un prototipo en un ambiente Telco 2.0, la composición de servicios convergentes basada en patrones del flujo de ejecución.

## 1.5 Alcance

El principal objetivo de este trabajo es soportar la composición de servicios convergentes mediante el uso de patrones de flujo de ejecución, los cuales se estudiaron, detectaron, implementaron y finalmente se validaron experimentalmente por medio de un prototipo funcional. La Figura 1-1 muestra un esquema de cinco módulos, el cual representa el proceso que se siguió para alcanzar dicho objetivo.

En el primer módulo, se explica el proceso de elaboración de un catálogo de servicios presentes en un entorno Telco 2.0, con el fin de establecer un banco de servicios sobre los cuales se pueda detectar patrones de flujo de ejecución. El segundo módulo, muestra la representación formal de los servicios del catálogo, lo cual permite homogenizar su descripción. En el tercer módulo, se utiliza un algoritmo para la detección de patrones de flujo de ejecución presentes en los servicios modelados. El cuarto, corresponde a la lista de patrones detectados por el módulo de detección. Finalmente, el quinto módulo explica la herramienta: **GT-4SC** (Graphical Tool-patterns-based for Service Creation) la cual permite componer servicios Telco 2.0 mediante los patrones de flujo de ejecución detectados.

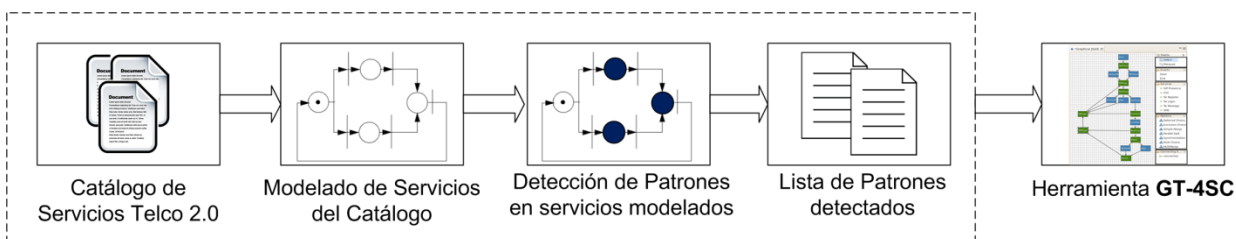


Figura 1-1: Alcance del trabajo

Por otra parte, la composición de servicios Telco 2.0 que se presenta en este trabajo, se hará de forma estática, la cual hace referencia a que el usuario escoge los servicios que desea componer en tiempo de diseño [22].

Adicionalmente, para el despliegue de servicios convergentes, existen variedad de plataformas, algunas de ellas como Rhino, son propietarias y su uso requiere el pago de licencias. Sin embargo, Mobicents tiene una licencia de libre uso, por lo cual fue la única plataforma utilizada para desarrollar el prototipo experimental del trabajo de grado.

Finalmente, se desarrolló un banco de pruebas con quince estudiantes de Ingeniería Electrónica y Telecomunicaciones de último semestre y estudiantes de primer año de la

maestría de Ingeniería Telemática. El objetivo de estas pruebas fue medir el tiempo de composición de tres servicios convergentes llamados: “TwitterCalls”, “TwitterSMS” y “CallMessage” haciendo uso de la herramienta **GT-4SC** y contrastar estos resultados con el tiempo empleado para la composición de los mismos en código fuente.

## 1.6 Contribuciones

- Un catálogo de servicios presentes en un entorno Telco 2.0.
- Un conjunto de patrones de flujo de ejecución que permiten componer servicios convergentes.
- Un prototipo experimental, donde se muestra el proceso de composición de servicios convergentes mediante patrones de forma gráfica.
- Un banco de pruebas el cual se desarrolló utilizando 15 estudiantes de Ingeniería Electrónica y Telecomunicaciones de último semestre, próximos a obtener su título y estudiantes de Maestría en Ingeniería Telemática, con conocimientos en programación.
- **Artículo:** “Control-Flow Patterns in Converged Services” presentado en la conferencia: “Service Computation 2012: The Fourth International Conferences on Advanced Service Computing” realizada en la ciudad de Niza, Francia.  
Artículo publicado en Agosto de 2012 con Copyright (c) IARIA, 2012. ISBN: 978-1-61208-215-8.

## 1.7 Estructura de la Monografía

El presente trabajo de grado se estructura de la siguiente manera: en el Capítulo 2 se estudian los conceptos principales y los trabajos relacionados con la temática de composición de servicios convergentes; el Capítulo 3 define un catálogo con la clasificación de servicios presentes en un entorno Telco 2.0; el Capítulo 4 presenta el proceso de detección de patrones de flujo de ejecución presentes en un conjunto de treinta servicios del catálogo, los cuales se modelaron mediante la representación formal de redes de Petri; el Capítulo 5 explica el prototipo experimental, el cual permite componer servicios convergentes de forma gráfica, mediante el soporte de patrones de flujo de ejecución; y finalmente, en el Capítulo 6 se presentan las conclusiones, resultados y trabajos futuros.

## Capítulo 2

### 2 Estado Actual del Conocimiento

Este Capítulo describe el estado actual del conocimiento alrededor de la composición de servicios convergentes. La Sección 2.1, presenta la base conceptual sobre la cual se centra la presente propuesta; y en la Sección 2.2, se exponen los principales trabajos relacionados.

#### 2.1 Base Conceptual

##### 2.1.1 Entorno Telco 2.0

Telco 2.0 es una nueva tendencia que busca reformular los modelos de negocio tradicionales de las telecomunicaciones. Este modelo surge por la necesidad de relacionar los conceptos, servicios y tecnologías de la Web 2.0 y el mundo Telco, donde se encuentran todos los operadores de telecomunicaciones y sus servicios tradicionales de telefonía [9].

Un entorno Telco 2.0 está constituido básicamente por dos capas: una capa superior llamada servicios o aplicación, y una capa de control. En la capa de aplicación se ubican servicios que pueden ser invocados siguiendo una lógica de implementación, para obtener servicios compuestos con funcionalidades adicionales que un solo servicio no posee. La capa de aplicación puede ser implementada haciendo uso de la especificación JAIN SLEE, la cual cumple con los requerimientos de rendimiento, y ofrece las condiciones necesarias para la ejecución de servicios convergentes. Por otra parte, la capa de control también puede ser implementada por JAIN SLEE, ya que a través del uso de adaptadores de recurso como: HTTP Servlet, Jain SIP, SMPP, Diameter, entre otros, es posible recibir la señalización que proviene de diferentes redes. Sin embargo, la arquitectura más usada para el control es IMS, la cual está basada completamente en tecnologías IP, y es el sistema más adecuado para realizar el control en el modelo de arquitectura de redes de nueva generación (NGN); esta capa está encargada del inter-funcionamiento de la red de transporte, con los servicios y aplicaciones de la capa superior [23].

En resumen, el principal objetivo de este enfoque de convergencia, es ofrecer al usuario servicios cada vez más completos, innovadores y con mayor funcionalidad, de tal forma que estos sean independientes de la infraestructura de red.

Para propósitos del presente trabajo de grado, se utilizó la especificación JAIN SLEE tanto en la capa de Aplicación como en la de Control, ya que el objetivo principal es soportar la composición de servicios, los cuales están alojados en la capa superior. La integración JAIN SLEE - IMS, se encuentra por fuera del alcance de este trabajo, y se propone como trabajo futuro.

### 2.1.2 Composición de servicios

Tanto el mundo de las Telecomunicaciones como Web, ofrecen sus funcionalidades en forma de servicios, los cuales son aplicaciones modulares y auto contenidas que pueden ser publicadas, localizadas e invocadas sobre la infraestructura de red. Sin embargo, cuando un servicio ya no cumple los requerimientos de los usuarios, debe existir la posibilidad de combinar servicios ya existentes con el fin de solucionar esta necesidad; es entonces donde surge el concepto de composición, en el cual un conjunto de servicios son combinados, siguiendo unas reglas de composición definidas, para obtener funcionalidad que un solo servicio no ofrece [2].

Existen diferentes esquemas de composición basados en la interacción entre servicios, las cuales son: composición estática, automática y dinámica. A continuación se profundiza sobre cada una de ellas.

- Composición estática: toma lugar durante el tiempo de diseño, cuando la arquitectura y el diseño del software son planeados. Los componentes a ser usados son escogidos, enlazados y finalmente compilados y desplegados. La composición estática, puede ser demasiado restrictiva cuando se requiere la adaptación de los componentes a cambios impredecibles [22].
- Composición automática: en este esquema de composición, los servicios son enlazados durante el tiempo de ejecución y el ambiente de servicio es altamente flexible y dinámico, es decir, la composición automática no se encarga de la ejecución de un servicio, sino de construir la síntesis de composición del servicio; este proceso está basado en los requerimientos solicitados y puede estar inmerso tanto en una composición estática como dinámica, ya que se encarga de crear el árbol de ejecución a partir de la coordinación de componentes del servicio, el cual es necesario para realizar todo mecanismo que permita la interacción de los servicios [24].
- Composición dinámica: se utiliza cuando es necesario mantener y mejorar la tolerancia a fallos de los servicios, en un entorno que posee condiciones altamente variables para el desarrollo y despliegue de los mismos. Este problema se puede contrarrestar con la composición dinámica, la cual se caracteriza por la capacidad de adaptarse, en tiempo de ejecución, a cambios inesperados que afecten el flujo de ejecución normal de un servicio y generar las soluciones adecuadas para garantizar su funcionamiento. Por lo tanto, se puede definir como un proceso de composición flexible, que en tiempo de ejecución brinda diferentes soluciones para mantener servicios en funcionamiento, capaz de responder a fallos que no fueron tenidos en cuenta o son imprevisibles en la fase de diseño [22].



Por otra parte, también se puede estudiar la composición cómo se presenta en el trabajo de Niemöller *et al.* [13], en el cual se explican tres enfoques: el primero es la composición de servicios en la Web; el segundo es la composición de servicios Telco; y el tercero es llamado composición unificada, o composición de servicios convergentes, la cual combina servicios de Telecomunicaciones y Web, de tal manera que al usuario final sean ofrecidos como un solo servicio.

Desde el punto de vista de la Web, se han realizado esfuerzos para lograr componer servicios mediante lenguajes bien definidos como BPEL, JPDL, WSCL (Web Services Conversation Language), BPML (Business Process Modeling Language) entre otros, los cuales permiten definir, en tiempo de diseño, el flujo de ejecución del servicio haciendo uso de patrones. Por otra parte, desde el mundo de las telecomunicaciones, las soluciones que se basan en FSM (Finite State Machine) han sido las más comunes ya que soportan la orientación a eventos de este tipo de servicios; sin embargo las únicas tres propuestas que han sido estandarizadas [25], son: La arquitectura IMS (IP Multimedia Subsystem) de la 3GPP, los servidores de aplicación que soportan la especificación JAIN SLEE y el API para Sip Servlets. Finalmente, desde el punto de vista de la composición unificada existen varias aproximaciones, algunas de ellas utilizan los lenguajes bien definidos propios de la Web para hacer invocaciones a servicios Telco y Web, como los trabajos [4] [5] [6]. Otros proyectos internacionales como OPUCE [19] y SPICE [20], los cuales se desarrollaron con ayuda de operadores de telecomunicaciones Europeos, han realizado importantes avances en la composición de servicios convergentes, utilizando APIs como la de Parlay X [26], la cual permite ofrecer los servicios de telecomunicaciones como WSDL (Web Services Description Language), para componerlos con servicios Web.

Sin embargo, a pesar de todos estos esfuerzos, aún existen problemas relacionados con la heterogeneidad de ambos mundos, la cual se debe a que el tiempo de entrega de un servicios Web a un usuario, tiene cierto grado aceptación, sin embargo en el entorno de las telecomunicaciones, algunos segundos de diferencia, implican la imposibilidad de comunicación entre usuarios.

### **2.1.3 Patrones de flujo de ejecución**

El flujo de ejecución, el cual integra el flujo de control de un determinado sistema con los datos que ingresan a este, es utilizado en muchos campos, por esta razón se habla de flujo de ejecución en áreas como: la Ingeniería de Software a la hora de desarrollar un programa; Ingeniería Automática, para determinar las entradas y salidas de un sistema; Procesos de Negocio (BP), para representar un conjunto de procedimientos, métodos y tareas en una organización; entre muchos otros. De este último surgen una gran cantidad de estudios de investigación, debido a que empresas de muchos sectores, buscan automatizar sus BP, para mejorar su productividad.

La necesidad de buscar estándares para la Gestión de BP (BPM) que permitan documentar, modelar, analizar, simular y ejecutar dichos BP, dio origen a muchos lenguajes como BPEL, JPDL, XPDL, entre otros, los cuales son capaces de representar y ejecutar el flujo de control y datos de los BP de forma automática o semi-automática. Sin embargo, esta necesidad también dio origen a un estudio que comenzó alrededor de 1999, liderado por el profesor Will

M.P van der Aalst, el cual definió un conjunto de patrones<sup>6</sup> titulados como: *Control-Flow Patterns* (CFP, Patrones de flujo de control) y *Data Patterns* (DP, Patrones de Datos), los cuales se definen en los trabajos [7] y [8]. Estos patrones permiten representar ambos flujos independientemente de la tecnología de implementación y al mismo tiempo presentan independencia de un dominio específico de aplicación.

Debido a su característica de independencia de uso en cualquier dominio, el conjunto de CFP y DP, se utilizó en el presente trabajo de grado, para la composición de servicios Telco 2.0, además, para mayor comprensión, se puede encontrar una descripción detallada de los mismos, basada en el estudio de Vander Aalst *et al.* en los Anexos A y B.

#### 2.1.4 Especificación JAIN SLEE

La especificación JAIN SLEE [17] define una arquitectura estándar de un entorno diseñado y optimizado para desarrollo de aplicaciones y servicios de telecomunicaciones haciendo uso del lenguaje java. Esta especificación, define un modelo de componentes para estructurar la lógica de las aplicaciones, como un grupo de componentes reutilizables orientados a objetos, con la posibilidad de reorganizarlos en servicios más complejos y de más alto nivel. La arquitectura de JAIN SLEE también define la forma cómo interactúan los componentes y sus contenedores en tiempo de ejecución.

JAIN SLEE provee un modelo de programación estándar que puede ser utilizado por la amplia comunidad de desarrolladores Java. El modelo de programación ha sido diseñado para simplificar el trabajo del desarrollador, eliminar los errores de programación y asegurar que servicios altamente robustos, puedan ser desarrollados en poco tiempo. Adicionalmente, el lenguaje Java facilita que las aplicaciones puedan ser desarrolladas una vez y luego desplegadas en cualquier entorno que implemente dicha especificación independientemente del SLEE [27].

El elemento fundamental definido por esta especificación es el SBB (Service Building Block). Los SBB son componentes software propios de aplicaciones asíncronas, que realizan actividades dependiendo de su estado actual. Estos componentes dependen de su estado debido a que son capaces de recordar los resultados de actividades previas, los cuales pueden ser aplicados en actividades adicionales.

Los SBB envían y reciben eventos, los cuales son usados para representar un acontecimiento que puede requerir procesamiento por parte de alguna aplicación, de esta forma, una aplicación orientada a eventos define métodos que son invocados cuando un evento es enviado, inspeccionado y procesado. Un evento se puede originar de diferentes fuentes, por ejemplo: una pila de protocolos externa, desde el interior del entorno SLEE, desde componentes de aplicación dentro del entorno SLEE o simplemente desde otros recursos basados en eventos.

Los recursos son entidades externas que interactúan con otros sistemas por fuera del entorno SLEE, tales como elementos de red (servidores de mensajería, servidores SIP, pilas de protocolos, directorios, bases de datos, etc.). Un adaptador de recursos adecua los

---

<sup>6</sup> Patrón: un patrón se define como: “una abstracción de una forma concreta que mantiene la aparición periódica en contextos específicos no arbitrarios”. Tomado de: [86]

requisitos y las interfaces particulares de un recurso a las interfaces y requisitos de JAIN SLEE [27].

### 2.1.5 Plataforma de entrega de servicios

El concepto de SDP (Service Delivery Platform) fue creado esencialmente para describir una solución para la entrega de servicios de valor agregado a consumidores y empresas. Dicha solución puede ser desplegada mediante proveedores de servicios de redes fijas y móviles. De esta manera una plataforma de entrega de servicios SDP tiene como objetivo principal permitir a los proveedores el rápido desarrollo y despliegue de nuevos servicios convergentes [28].

La SDP es un modelo basado en estándares, que generalmente proporciona un entorno de creación y composición de servicios, un entorno de ejecución y gestión de servicios, y abstracciones para control de medios, presencia, localización y otras funciones para comunicaciones a bajo nivel con el fin de lograr una integración de los servicios de telecomunicaciones con los servicios de internet.

Los componentes básicos de una SDP se describen a continuación [28].

- **Entorno de creación de servicios (SCE):** el SCE es el principal componente para los desarrolladores de servicios de telecomunicaciones, ya que éste es utilizado para crear el software de los servicios desplegados. En un SCE se lleva a cabo la construcción paso a paso de los servicios, describiendo los elementos e interacciones entre sí que conforman el ciclo de vida del servicio; además el SCE cuenta con una variedad de componentes que permiten la reutilización de servicios y por lo tanto la composición de estos, con el fin de obtener robustos procesos de negocio.
- **Entorno de ejecución de servicios (SEE):** el SEE es el componente de la plataforma que soporta la ejecución de los servicios que han sido creados a través del SCE, cumpliendo con ciertos requerimientos de escalabilidad, baja latencia y tolerancia a fallos. Para dicha tarea, este entorno usualmente contiene una pasarela de eventos y un motor de ejecución de la lógica del servicio, que permite realizar la composición descrita en el SCE.
- **Habilitadores:** los Habilitadores son bloques de construcción que exponen una simple API, la cual permite al desarrollador abstraer la complejidad de las funciones de la red subyacente. Una SDP puede hacer uso de Habilitadores que se encuentren tanto dentro de la plataforma como fuera de ella, para lograr servicios que utilicen variedad de funciones. Usualmente se utilizan Habilitadores para control de llamadas, gestión de perfiles de usuario, presencia, localización, mensajería, control de medios, aplicaciones SIP, e integración con sistemas OSS/BSS.

Algunas de las plataformas que contienen los componentes necesarios para el despliegue de servicios se describen a continuación.

**Plataforma OpenCloud Rhino:** El entorno de ejecución de lógica de servicios (SLEE) Rhino es un servidor de aplicaciones que permite el desarrollo de servicios convergentes. Es una

plataforma Java que soporta las especificaciones JAIN SLEE 1.0 (JSR 22) y JAIN SLEE 1.1 (JSR 240) [17].

El SLEE Rhino puede ser usado para desarrollar y desplegar aplicaciones que usen los protocolos basados en SS7 tales como INAP y CAP, protocolos IMS como ISC y Diameter, y protocolos basados en IP como HTTP y SIP. Rhino tiene una infraestructura dirigida a los operadores de telecomunicaciones y se caracteriza por ser tolerante a fallos, por proveer una disponibilidad continua y por soportar una gestión en línea.

**Plataforma de comunicaciones de Mobicents:** Es un servidor orientado a eventos altamente escalable con un modelo de componentes robusto y un entorno de ejecución con tolerancia a fallos.

La arquitectura de la plataforma de comunicaciones Mobicents se encuentra diseñada para la creación, despliegue y gestión de servicios y aplicaciones que integran voz, video y datos a través de redes IP y de comunicaciones. Mobicents logra la convergencia aprovechando las siguientes capacidades centrales [3]:

- **JAIN SLEE:** Mobicents JAIN SLEE es una implementación certificada de la especificación JAIN SLEE v1.1 (JSR 240) de libre distribución. Mobicents brinda un modelo de componentes y un entorno de ejecución robusto para aplicaciones de telecomunicaciones. Es un complemento para J2EE que permite la convergencia de voz, video y datos en aplicaciones inteligentes de nueva generación.
- **SIP Servlets:** Mobicents Sip Servlets entrega una plataforma abierta y consistente, en la cual desarrollar y desplegar servicios SIP y JEE portables y convergentes. Es la primera implementación certificada de la especificación SIP Servlet v1.1 (JSR 289) de libre distribución que trabaja sobre contenedores Tomcat y JBoss.
- **Media Server:** Mobicents Media Server es un servidor de libre distribución basado en la implementación Java Media Framework cuyo objetivo principal es proveer una funcionalidad completa de una pasarela de contenido de la más alta calidad. El servidor satisface las necesidades de redes inalámbricas, cableadas y de redes fijas y móviles IP convergentes desde una única plataforma de contenido.
- **JBoss Microcontainer:** JBoss Microcontainer es el entorno de almacenamiento en el cual residen los contenedores de más alto nivel. Provee servicios de registro, configuración, gestión, control de utilización de clases, empaquetamiento, despliegue y muchos otros bloques necesarios en servidores altamente escalables y tolerantes a fallos.

En el ámbito de las telecomunicaciones y de las redes de nueva generación, Mobicents se presenta como un núcleo de alto rendimiento para plataformas de entrega de servicios (SDP). Mobicents permite la composición de bloques constructores de servicio (SBB) tales como control de llamada, facturación, aprovisionamiento de usuario, administración y capacidades de detección de presencia.

**Ericsson SDP:** es un conjunto de componentes que permiten a los operadores entregar nuevos servicios a través de una red horizontal de servicios [29]. Esta SDP tiene como objetivo soportar una fuerte integración con las operaciones y procesos de negocio de los

operadores, además de proporcionar la posibilidad de gestionar completamente el ciclo de vida de los servicios, incluyendo la creación, despliegue, aprovisionamiento, finalización, facturación, operación y mantenimiento.

Esta SDP está basada en el principio de reutilización de funcionalidad de SOA, por lo tanto define componentes que permiten agrupar funciones comunes de los servicios. Estos componentes son: el entorno de creación de servicios (SCE), servidores de aplicaciones, los habilitadores de servicios y un componente de soporte de funciones comunes

Finalmente, para el presente trabajo de grado, la plataforma escogida para el despliegue de servicios fue Mobicents ya que es la única SDP de libre distribución, acceso y descarga, que implementa la especificación JAIN SLEE, lo que permite su uso para el despliegue de servicios sin ningún tipo de licencia.

## 2.2 Trabajos Relacionados

En la Sección 2.1 se han enumerado algunos trabajos, los cuales hacen un importante aporte al trabajo de grado, como los patrones de flujo de ejecución [7] [8] [30], la definición de un entorno Telco 2.0 [9] [23], la especificación JAIN SLEE [17] [3], las plataformas para la creación de servicios [31] [32], entre otros. Sin embargo, los trabajos que se presentan a continuación se relacionan muy estrechamente al trabajo de grado, ya que plantean el mismo problema de componer y desplegar servicios de una manera más simple y en el menor tiempo posible. Estos trabajos se describen a continuación.

**A software architecture for simplifying the JSLEE service design and creation:** este trabajo desarrollado por Femminella et al. [5], plantea una nueva arquitectura basada en el entorno JAIN SLEE [17], la cual permite separar la lógica de negocio del servicio, de su implementación, con el objetivo de facilitar su gestión y desarrollo.

Cuando no se cuenta con los conocimientos técnicos apropiados, el desarrollo y la composición de servicios puede convertirse en un proceso ineficiente, por esta razón, este trabajo muestra una solución basada completamente en software libre, la cual se enfoca en la integración de un motor de Workflow dentro de la plataforma JAIN SLEE, de tal forma que la creación de un servicio puede ser abstraída como un proceso de negocio y diseñada gráficamente mediante un conjunto de bloques unidos por transiciones.

Para la implementación del entorno de ejecución, este proyecto utiliza dos módulos. El primero es la plataforma JAIN SLEE, donde está la lógica y el flujo de ejecución del servicio, a través de SBB. El segundo módulo, utiliza la herramienta JBPM como sistema de gestión del flujo de trabajo (WFMS), la cual trae incorporado un lenguaje nativo de flujo de trabajo llamado JPDL. JBPM ofrece una fuerte integración con entornos java y soporta de forma nativa el servidor de aplicaciones JBOSS. Además, JBPM es una herramienta gráfica que soporta el uso de patrones como *Multi-Choice*, *Synchronization Sequence*, entre otros. Esta herramienta actúa como un intermediario entre los diseñadores de procesos de negocio y los desarrolladores, ya que permite la combinación de bloques sin la necesidad de dominar el lenguaje de descripción de cada proceso.

La validación de la solución propuesta por este proyecto, se realiza mediante la creación de un servicio compuesto de telecomunicaciones, el cual permite hacer llamadas internacionales mediante el uso de tarjetas prepagadas.

**Enhanced concept of the TeamCom SCE for automated generated services based on JSLEE:** este trabajo fue desarrollado por Eichelmann *et al.* [4], el cual está enmarcado en el proyecto TeamCom, el cual ofrece a los desarrolladores, una forma fácil, rápida y rentable para crear servicios de telecomunicaciones, mediante el uso de una herramienta gráfica BPEL, la cual permite el uso de patrones como *Secuence, If, Flow, While*, entre otros. Dicha herramienta hace uso de un generador de código, el cual traduce la descripción lógica del servicio en BPEL a un código en lenguaje java, para posteriormente desplegarlo en un servidor de aplicaciones JAIN SLEE.

El generador de código analiza paso a paso cada uno de los procesos y tareas del flujo de trabajo descrito en BPEL, y a partir de dicha información, genera archivos correspondientes a cada elemento como: clases java y descriptores XML, necesarios para la implementación del servicio JAIN SLEE.

Este proyecto presenta una arquitectura claramente definida la cual incluye un ambiente de creación, una capa para el despliegue de servicios, y un ambiente de ejecución el cual contiene uno o varios servidores de aplicación basados en JAIN SLEE. Adicionalmente, se define la capa de transporte de servicio, la cual permite abstraer diferentes protocolos con el fin de brindar independencia a las capas superiores, todo esto con el fin de facilitar el proceso de creación, despliegue y ejecución de servicios.

Este artículo especifica ocho bloques constructores elementales llamados: bloques constructores de comunicación (CBB), estos se derivan de los requerimientos de las telecomunicaciones y permiten crear servicios mediante su combinación. Dicho conjunto de CBB está compuesto por los siguientes servicios: entrada de datos, salida de datos, activación de datos, archivos, video; texto, conferencia y audio.

**HSCEE: a highly flexible environment for hybrid service creation and execution in converged networks:** este trabajo desarrollado por Zhu *et al.* [6], propone una arquitectura mediante un entorno de creación y ejecución híbrido de servicios (HSCEE), el cual permite crear servicios convergentes gráficamente. Esta arquitectura está basada en la implementación de la especificación JAIN SLEE, el lenguaje BPEL y un bus de servicios empresariales (ESB).

Este proyecto define una arquitectura que es dividida en tres capas principales, las cuales se explican a continuación:

- Capa de creación de servicios (SCL): permite crear un modelo gráfico-abstracto del servicio, a través de plantillas (templates) BPEL. Lo más sobresaliente de esta capa, es la capacidad de crear nuevos modelos de servicios, a través de otros previamente realizados. Cuando el modelo abstracto ha sido creado, esta capa se encarga de transformar este modelo a un modelo de servicio, para que sea ejecutado por la capa de ejecución del servicio (SEL).
- Capa de ejecución del servicio (SEL): es la encargada de la ejecución de los servicios Web y de telecomunicaciones. Emplea 3 bloques principales, el primero

es llamado Motor dinámico de ejecución de procesos (DPEE), el cual se encarga de ejecutar los servicios Web, el segundo es JAIN SLEE encargado de la ejecución de servicios de telecomunicaciones, y manejar los protocolos que funcionan mediante invocaciones asíncronas; y por último, un bus de servicios empresariales (ESB) se encarga de intermediar el tráfico entre el DPEE y JAIN SLEE para ejecutar los servicios.

- Capa de abstracción de recursos (SAL): esta capa representa la utilización de diferentes tipos de recursos como: bases de datos, dispositivos de red, directorios, mensajería SIP, entre otros.

Para validar esta propuesta, el grupo de investigación implementó un servicio de conferencia en el cual los usuarios pueden acceder a través de un Softphone, teléfono móvil o desde un teléfono fijo, y unirse a la conversación.

**Proyecto SPICE:** (Service Platform Innovative Communication Enviroment), es un proyecto de investigación, sobre el cual se hicieron aproximadamente 70<sup>7</sup> publicaciones. Tiene como objetivo principal, promover la diseminación de servicios móviles personalizados y eficientes, permitiendo a los desarrolladores, definir servicios enriquecidos semánticamente. Dentro de este proyecto se definió un lenguaje de programación llamado SPATEL [20], el cual permite la definición de servicios ejecutables semánticamente enriquecidos.

En el proyecto SPICE se definen dos entornos principales:

- Un entorno de creación de servicios (SCE), el cual contiene herramientas para los desarrolladores que permiten la creación y composición de servicios.
- Un entorno de ejecución de servicios (SEE), el cual incluye un servidor de aplicaciones, donde se despliegan los servicios y también dispone de una infraestructura de nivel inferior.

La Universidad de Twente contribuyó en el proyecto con el desarrollo de un motor de composición automática (ACE), el cual es el componente clave del SCE. El objetivo principal del ACE es permitir la composición dinámica de servicios. Uno de los bloques de construcción principales del ACE es el algoritmo de composición, el cual automáticamente realiza la composición de servicios a partir de servicios previamente desarrollados y desplegados en un entorno de comunicaciones, de acuerdo con una petición semántica específica realizada por un usuario final.

**Proyecto OPUCE<sup>8</sup>:** (Open Platform for User-Centric Service Creation and Execution), este proyecto de investigación fue financiado por la Unión Europea y tuvo como objetivo, la creación de una plataforma en la cual usuarios inexpertos, podrían crear y publicar sus propios servicios.

En [19] se hace una breve descripción de los principales componentes de plataforma OPUCE, la cual abarca todos los elementos del ciclo de vida de un servicio, como creación, despliegue, ejecución, etc. Los componentes principales se explican a continuación:

<sup>7</sup> <http://www.ist-spice.org/nav/publications.htm>

<sup>8</sup> <http://www.opuce.eu/>

- Editor de servicios avanzado: es un editor Web gráfico en el cual los usuarios pueden componer, desplegar servicios convergentes, además permite la gestión de servicios previamente creados.
- Gestión del ciclo de vida del servicio: maneja el ciclo de vida del servicio: despliegue, aprovisionamiento, y monitoreo. Este módulo también es responsable de transformar los servicios – OPUCE – compuestos, creados por los usuarios, en scripts BPEL.
- Entorno de ejecución del servicio: encargado de almacenar y ejecutar el código de los servicios. Utiliza tecnologías como JAIN SLEE .Net y J2EE.
- Manejo de la información del usuario: almacena la información del usuario como: perfil, contexto (ubicación, presencia, capacidades del dispositivo, etc.).
- Recomendación de servicio: recomienda a los usuarios finales los servicios que mejor se acomoden al perfil de usuario.
- Repositorio de servicios: almacena la descripción de los servicios - OPUCE y los servicios de los usuarios según el perfil. Este repositorio incluye una poderosa capacidad de búsqueda basada en palabras clave y semántica.
- Recomendación según el contexto: permite la adaptación dinámica de los servicios de acuerdo a la información recibida del contexto y de los perfiles de los usuarios.

### Brechas de conocimiento

La Tabla 2-1 describe las brechas del conocimiento identificadas en los anteriores trabajos.

Tabla 2-1: Brechas del conocimiento

Trabajo	Brecha del conocimiento
[4], [5], [6]	<ul style="list-style-type: none"> <li>• Estos trabajos utilizan algunos patrones de flujo de ejecución descritos por los lenguajes BPEL o JPD, lo que facilita la composición de servicios convergentes. Sin embargo, es necesario incluir capas adicionales de transformación, para que estos patrones puedan ser ejecutados en JAIN SLEE.</li> <li>• El desarrollo de estos trabajos se encuentran enmarcados dentro de proyectos de investigación, por lo anterior, no es posible tener acceso a los prototipos software desarrollados.</li> </ul>
[4]	Brinda a los desarrolladores una forma fácil, rápida y rentable para crear servicios convergentes. Ofrece ocho servicios básicos y utiliza un generador de código el cual traduce la descripción lógica del servicio a un código en lenguaje java. Sin embargo, no se permite agregar servicios adicionales lo cual genera limitantes en términos de escalabilidad.
[19]	El principal componente de la arquitectura OPUCE, es un motor BPEL el cual se encarga de la lógica de ejecución de los servicios creados por el usuario. Esto presenta un limitante en términos de rendimiento, ya que este motor adiciona tiempo procesamiento, lo que puede causar retardos en la entrega del servicio.



[20]	El lenguaje SPATEL facilita la composición de servicios a usuarios expertos como también a usuarios inexpertos, sin embargo, no tiene soporte de patrones de flujo de ejecución y es un lenguaje propietario al cual no se puede tener acceso.
[5]	La principal desventaja de este trabajo es la integración del motor de Workflow externo al SLEE para ejecutar la lógica del servicio, ya que es una capa adicional de procesamiento, lo que puede generar problemas en términos de rendimiento. Por otra parte, los patrones de flujo de ejecución están limitados a la implementación de JBPM, lo cual no hace posible su ejecución directa en JAIN SLEE.
[6]	Las principales desventajas de este trabajo se dan en términos de rendimiento, ya que la integración de componentes adicionales al SLEE, como la capa de transformación de plantillas (templates) BPEL a código ejecutable, y el bus de servicios empresariales, adicionan tiempo de procesamiento. Lo anterior causa retrasos en la entrega del servicio, cuando el número de usuarios que simultáneamente hacen uso del servicio es considerable.

## 2.3 Resumen

Este Capítulo presentó el estado actual del conocimiento, sobre el cual se encuentra enmarcado el trabajo de grado, de la siguiente manera: se ilustra la nueva tendencia para la creación de servicios denominada Telco 2.0; se describió la composición de servicios de Telecomunicaciones, Web y convergentes; se mostraron los trabajos que definen los patrones de flujo de ejecución, se nombraron plataformas para creación y ejecución que brindan soporte para la composición; se describió la especificación JAIN SLEE; y finalmente se describieron los trabajos relacionados que integran todos los aspectos nombrados.



## Capítulo 3

### 3 Clasificación de Servicios

En la actualidad los Telcos entienden la necesidad de crear nuevos servicios de una manera más rápida y a un costo más bajo, con el fin de atraer nuevos clientes, mantener los existentes y aumentar sus ingresos. Por esta razón, es de vital importancia tener un conocimiento claro acerca de cuantos y cuales servicios están presentes en los entornos actuales y más aún, de que formas se pueden componer para crear nuevos servicios.

Este Capítulo presenta en la Sección 3.1, una serie de definiciones generales; la Sección 3.2 una revisión bibliográfica de servicios estandarizados por diferentes organizaciones como: ITU, 3GPP, ETSI-TISPAN, OneApi; algunas plataformas para el despliegue de servicios como: Mobicents, Rhino Slee, Ericsson SDP; y servicios ofrecidos por operadores de telecomunicaciones; la Sección 3.3, describe un modelo en capas de un entorno Telco 2.0 y se propone una clasificación de servicios basada en la revisión anterior; finalmente, en la sección 3.4, se presenta el catálogo de servicios propuesto, teniendo en cuenta la revisión bibliográfica y el modelo en capas del entorno Telco 2.0.

#### 3.1 Definiciones

Tanto en el mundo de las telecomunicaciones como de la Web existen definiciones y conceptos generales de gran importancia, relacionados con el tipo de servicios que cada uno de estos mundos prestan de forma individual o unificada, estos a su vez, han sido base para la definición de arquitecturas y clasificaciones de servicios que describen las características y funcionamiento de los mismos.

Algunos de estos conceptos y definiciones generales que abarcan ambos mundos son:

**Servicio:** un servicio es un grupo de funciones y facilidades desarrolladas y entregadas como un producto, el cual es suministrado por un proveedor a un usuario, en respuesta a su demanda, deseos o necesidades. Un mismo servicio puede ser usado como componente en varios productos, empaquetado de forma diferente y ofertado a distintos costos [33] [34].

**Servicio compuesto:** servicio cuya implementación realiza invocaciones a otro(s) servicio(s) de tal manera que cada uno de sus componentes pueden ser conectados entre sí para construir nuevos servicios.

**Servicio de valor agregado (VAS):** hasta el momento las organizaciones de regulación y estandarización internacional de telecomunicaciones no han expuesto una definición general de VAS que posibilite un entendimiento común de dicho término a nivel mundial. Por el contrario, el concepto de VAS actualmente depende del contexto regulatorio y de mercado de cada país (e.g: Colombia [35], Argentina [36], etc. ). A pesar de esto, la mayor parte de las instituciones exponen una descripción específica de VAS basándose en un mismo punto de vista: distinguir ciertos servicios avanzados de los servicios tradicionales, los cuales agregan otras facilidades diferenciables del servicio soporte satisfaciendo nuevas necesidades específicas de telecomunicaciones.

**Servicio convergente:** coordinación de un conjunto de servicios proporcionados mediante diferentes redes, tales que a la vista del usuario final sean un solo servicio [37].

**Servicio Telco 2.0:** servicio convergente que utiliza las capacidades del mundo Telco y de la Web 2.0. Incluye características como interactividad, ubicuidad, orientación y participación social o generación de contenido, propias de la Web 2.0; con servicios, de las telecomunicaciones como SMS, llamada, etc. [23].

### 3.1.1 Definiciones Telco

**Servicios de Telecomunicaciones:** conjunto de funciones independientes que son parte integral de los procesos de negocio, que satisfacen las necesidades específicas de telecomunicaciones de los usuarios [34], [38]. Están integrados por componentes hardware, software y de un medio de comunicación que constituyen un conjunto específico de capacidades para la transmisión de información a un grupo determinado de usuarios. Los servicios de telecomunicaciones son prestados por personas jurídicas, públicas o privadas, debidamente constituidas, con o sin ánimo de lucro, también llamadas proveedores de servicios de telecomunicaciones [39].

Los servicios de telecomunicaciones se clasifican en servicios básicos y portadores como se muestra en la Figura 3-1, los cuales corresponden a los servicios recibidos por el usuario final y servicios prestados por la infraestructura de los operadores respectivamente. Estos pueden ser complementados por capacidades adicionales o servicios suplementarios con el fin de brindar nuevas o mejores funcionalidades.

Servicios de Telecomunicaciones			
Servicios Portadores		Servicios Básicos	
S. Portadores Básicos	S. Portadores Básicos + Servicios Suplementarios	Servicios Básicos	Servicios Básicos + Servicios Suplementarios

Figura 3-1: Clasificación de servicios de telecomunicaciones; tomado de [40]

**Servicios Portadores:** un servicio portador es aquel que ofrece la capacidad necesaria para el transporte y enrutamiento de señales entre dos o más puntos de terminación o de interfaz usuario-red, ya sea en redes no conmutadas o de conmutación de circuitos o paquetes [41] [38].

**Servicio Básico:** término que hace referencia a los servicios tradicionales, es decir, aquellos que comúnmente son suministrados al usuario final por parte de un proveedor de telecomunicaciones [38]. Este tipo de servicio es también conocido como servicio final ya que proporciona en sí mismo la funcionalidad completa, incluida la del equipo terminal, para la comunicación entre usuarios a través de una red fija o móvil de acuerdo a los protocolos establecidos entre los operadores. Un servicio básico constituye la base para los servicios suplementarios [41] [42].

**Servicio Suplementario:** término con el cual se define al tipo de servicio que modifica o complementa un servicio básico. Se encuentra soportado sobre un servicio básico, razón por la cual no puede ser ofrecido como un servicio único. Un mismo servicio suplementario puede ser aplicado a varios servicios básicos de telecomunicaciones [38]. Comprenden funciones como llamada abreviada, identificación de llamada entrante, conferencia entre varios usuarios, etc.

En la Figura 3-2 se observa la ubicación de los servicios portadores y básicos dentro de un entorno de telecomunicaciones, estos a su vez pueden estar acompañados de uno o varios servicios suplementarios con el fin de ofrecer funcionalidades adicionales. De esta forma un servicio de telecomunicaciones está conformado por servicios que abstraen las capacidades de la red y un conjunto de aplicaciones (incluyendo las capacidades de terminal de usuario) que constituyen los servicios presentados al usuario final.

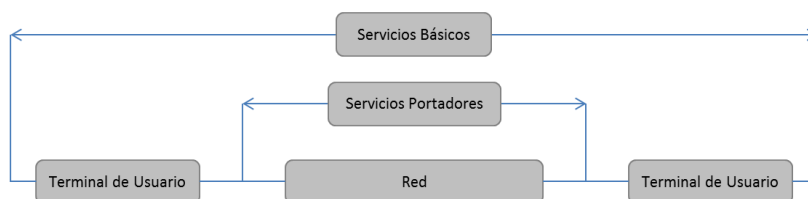


Figura 3-2: Ubicación de servicios portadores y básicos [43] [44]

**Servicios Compuestos Telco:** surgen de la unión o composición de servicios con funcionalidades específicas o básicas, de forma que estos cumplan un objetivo común para la creación de nuevos servicios de telecomunicaciones con funcionalidades nuevas o adicionales a los servicios tradicionales. Por otra parte la adición de servicios suplementarios a los servicios básicos resulta en servicios compuestos que ofrecen un valor agregado al usuario y permite una mayor oferta de servicios por parte de los Telcos.

### 3.1.2 Definiciones Web

**Servicios Web:** sistema software con interfaces y conexiones públicas definidas y descritas en XML. Puede ser descubierto por otros servicios a través de un Identificador de Recurso Uniforme (URI, Uniform Resource Identifier) e interactuar entre ellos de una manera prescrita por su definición y usando mensajes basados en XML transmitidos por protocolos de Internet [34].

**Servicios Web 2.0:** Web 2.0 es una plataforma de red en la que los usuarios contribuyen al desarrollo de herramientas, contenido y comunidades en Internet. Un Servicio Web 2.0 se distingue de las páginas y servicios Web tradicionales estáticos y pasivos, como paradigma de creación de conocimiento dinámico e interactivo en Internet [45].

En la actualidad se encuentran una incontable cantidad de servicios implementados en la web, por esto es necesario abstraer características comunes de estos para lograr su clasificación, en este sentido se han adelantado trabajos que clasifican servicios por: la tecnología que utilizan (SOAP, Rest), la funcionalidad que cumplen para el usuario [46], el grado de control sobre la información que tiene el sitio web, etc. Esta última es analizado en el trabajo de Shang *et al.* [45] desde la perspectiva: Knowledge-Creation (Creación de Conocimiento), el cual se basa en un ciclo, como se muestra en la Figura 3-3, que permite la divulgación de información para que los usuarios se apropien de esta, permitiendo así, la creación de nuevo conocimiento.

La primera fase de este ciclo es llamada: socialización, donde los usuarios: comparten información en la Web 2.0, participan en comunidades, se comunican mediante VoIP, etc. La segunda fase es llamada: externalización, donde los usuarios escriben mails, comparten información mediante chat, etc. La tercera fase es llamada: combinación, donde se emplean tecnologías de la web 2.0 para compartir la información, como: RSS (Really Simple Syndication), Etiquetado (Tagging), Folksonomias, etc. Por último, la internalización es un proceso sistemático que refleja el aprendizaje colectivo a través de acciones y prácticas.



Figura 3-3: Ciclo del modelo de creación de conocimiento en sitios Web 2.0  
Figura tomada de [45]

Las 4 fases en el modelo de creación de conocimiento SECI (Socialización S, Externalización E, Combinación C, Internalización I) son tenidas en cuenta para la clasificación de los servicios y adicionalmente el grado de control que tiene el servicio sobre la información y el servicio al cliente; estos últimos se explican a continuación.

- El grado de control que tiene el servicio sobre la información que se comparte a través de éste, el cual es definido como bajo y alto. Bajo: cuando existe un registro de los usuarios, la información compartida es estandarizada. Alto: cuando la información compartida tiene que ser revisada y autorizada para que sea compartida con otros usuarios.

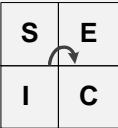

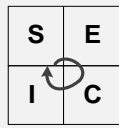
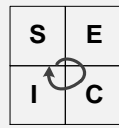
- El servicio al cliente: el proceso de aprendizaje del usuario se puede beneficiar a través de las características de la WEB 2.0, como: la asimilación, regeneración, reinterpretación y la retención del nuevo conocimiento.

Teniendo en cuenta las fases del modelo (SECI), el grado de control que tiene el servicio sobre la información y el valor que le da el usuario al servicio, se completa el modelo para la clasificación de los servicios, como se muestra en la Tabla 3-1, definiendo así cuatro roles principalmente:

- Intercambiador: Comprende S – E, con mecanismo de control bajo (Tipo I).
- Agregador: Comprende S – E – C, con mecanismo de control bajo (Tipo II).
- Colaborador: Comprende S – E – C – I, con mecanismo de control alto (Tipo III).
- Liberador: Comprende S – E – C – I con mecanismo de control Bajo (Tipo IV).

Es así como los servicios de la Web 2.0 son clasificados teniendo en cuenta los roles anteriormente descritos, algunos ejemplos se muestran en la Tabla 3-1.

Tabla 3-1 Clasificación de servicios de la WEB 2.0, siguiendo el modelo de creación de conocimiento

	<b>Intercambiador</b>	<b>Agregador</b>	<b>Colaborador</b>	<b>Liberador</b>
Aplicaciones Web 2.0	Llamadas y VoiP, Chat, Email	Blogs, Música, foto, video RSS	Wikis, oficina, negocios, programación, juegos	Negocios, oficina, programación
Tipo	I	II	III	IV
Circulo de Creación de Conocimiento.				
Mecanismo de Control	Registro	Registro, estandarización	Registro, estandarización, sistematización, autorización, revisión	Acceso, estandarización
Servicio para el cliente	Compartir información instantáneamente	<ul style="list-style-type: none"> <li>• Compartir y retener información</li> <li>• Información suficiente</li> </ul>	<ul style="list-style-type: none"> <li>• Compartir, retener, asimilar conocimiento bajo estándares específicos</li> <li>• Conocimiento de calidad</li> </ul>	<ul style="list-style-type: none"> <li>• Compartir, retener, asimilar conocimiento bajo estándares libres</li> <li>• Conocimiento de libre acceso</li> </ul>
Ejemplos	MSN, Skype	Twitter, Youtube, Facebook	Wikipedia, Answers.com	Linux

**Servicios Web compuestos:** es un servicio Web cuyas operaciones son interfaces a operaciones proporcionadas por otros servicios Web, también conocidos como Servicios Web Atómicos. Se debe resaltar que cada Servicio Web Atómico puede proporcionar solo un subconjunto de las operaciones requeridas por el Servicio Web Compuesto.

En general, cuando una aplicación cliente invoca operaciones de un servicio Web Compuesto, éste no ejecuta esas operaciones por sí mismo, en cambio, se comporta como un mediador que delega las operaciones a algunos servicios Web atómicos. La Figura 3-4

muestra un escenario de ejemplo de invocación de operaciones entre la aplicación cliente, el servicio Web Compuesto y los servicios Web Atómicos. Cuando la operación A en el servicio Compuesto es invocada por la aplicación cliente, tanto **WS1.A** o **WS3.A** pueden ser seleccionados para la ejecución.

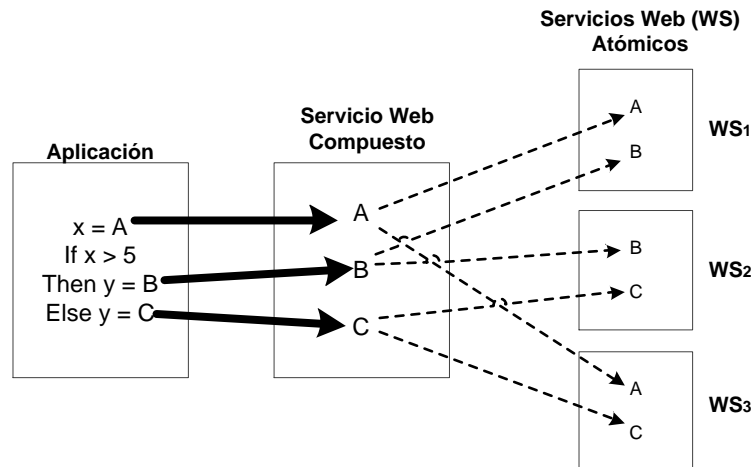


Figura 3-4: Servicio Web compuesto

## 3.2 Revisión de servicios de Telecomunicaciones, Web y Convergentes

A continuación, en la Sección 3.2.1 se presentan un conjunto de servicios estandarizados y clasificados por Organizaciones como la 3GPP, ITU, ETSI-TISPAN, ONEAPI y TelcoML; en la Sección 3.2.2 se hace una revisión de los servicios que están integrados a SDP's como Mobicents, Rhino y la SDP de Ericsson; y finalmente en la Sección 3.2.3, se describen los servicios ofrecidos por algunos operadores de telecomunicaciones

### 3.2.1 Organizaciones de estandarización

Las organizaciones de estandarización son organismos encargados de establecer los diferentes estándares utilizados a nivel mundial en diferentes áreas: telecomunicaciones, redes, sistemas móviles, etc. Dichos estándares corresponden a un conjunto de normas y recomendaciones técnicas que regulan la transmisión en los sistemas de comunicaciones.

Existe una variedad muy grande de organizaciones de estandarización en el mundo, a continuación se presentan algunas de ellas.

**3GPP:** El Proyecto Asociación de Tercera Generación (3rd Generation Partnership Project), es una colaboración de grupos de asociaciones de telecomunicaciones para la producción de informes y especificaciones que definen las tecnologías de esta organización las cuales han estado en constante evolución a través de lo que se conoce como generaciones de sistemas comerciales móviles /celulares [47].



Las especificaciones del proyecto 3GPP se estructuran como versiones o releases, las cuales pueden tener diferentes revisiones. Cada versión incorpora centenares de estándares individuales que proporcionan una visión completa y detallada de cómo funciona la industria de las telecomunicaciones móviles y proporcionan a los desarrolladores una plataforma de implementación que permite la adición de nuevas características requeridas por el mercado.

Dentro de las especificaciones podemos encontrar algunas relacionadas con la clasificación de servicios de telecomunicaciones y con algunos estándares, herramientas y APIs que describen su funcionamiento y dan una visión clara acerca de su implementación:

Algunas de estas especificaciones son:

- 3GPP TS 22.002: servicios Portadores soportados por una red PLMN.
- 3GPP TS 22.003: servicios Básicos soportados por una red PLMN.
- 3GPP TS 22.101: principios y aspectos generales de Servicios soportados por una red PLMN.
- 3GPP TS 22.004: aspectos generales de servicios suplementarios.
- 3GPP TS 29.199-[1-22]: servicios Web Parlay X.

**ITU:** La Unión Internacional de Telecomunicaciones (ITU, International Telecommunication Union) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones, y fue creado el 17 de mayo de 1865 como una organización intergubernamental en la cual los Estados miembros y el sector privado de las telecomunicaciones coordinan el desarrollo y operabilidad de las redes y servicios de telecomunicaciones [48].

La ITU es responsable de la regulación, normalización y desarrollo de las telecomunicaciones a nivel mundial, al tiempo que vela por la armonización de las políticas nacionales de telecomunicaciones. Forman parte de la UIT 193 Estados Miembros y varios centenares asociados.

La ITU está conformada por tres sectores:

- ITU-T: sector de normalización de las telecomunicaciones: estudia los aspectos técnicos, de explotación y tarifarios y publica normativa sobre los mismos, con vista a la normalización de las telecomunicaciones a nivel mundial.
- ITU-R: sector de normalización de las radiocomunicaciones: su función es función es regular la mayor parte del espectro radioeléctrico.
- ITU-D: sector de desarrollo de las telecomunicaciones de la ITU: creado para contribuir a difundir un acceso equitativo, sostenible y asequible a las telecomunicaciones, y de este modo, fomentar un mayor desarrollo económico y social.

En general, la normativa generada por la ITU está contenida en un amplio conjunto de documentos denominados: recomendaciones, agrupados por series. Un ejemplo de estas

son las recomendaciones de la ITU-T de la Serie A hasta la Serie Z [49], algunas de las cuales son:

- Serie F: servicios de telecomunicaciones no telefónicos. Algunas recomendaciones se muestran en la Tabla 3-2.

Tabla 3-2: Recomendaciones de la serie F de la ITU

Recomendación	Descripción
F.160-F.399	Servicios telemáticos.
F.400-F.499	Servicios de manejo de mensajes.
F.600-F.699	Servicios de transmisión de datos.
F.700-F.799	Servicios audiovisuales.

- Serie I: red digital de servicios integrados. Algunas recomendaciones se muestran en la Tabla 3-3.

Tabla 3-3: Recomendaciones de la serie I de la ITU

Recomendación	Descripción
I.112	Vocabulario de términos relativos a las redes digitales de servicios integrado.
I.220	Descripción dinámica de los servicios de telecomunicación básicos.
I.230	Definición de las categorías de servicios portadores.
I.240	Definición de teleservicios
I.250	Definición de servicios suplementarios.

- Serie Y: infraestructura mundial de la información, aspectos del protocolo internet y redes de la próxima generación. Algunas recomendaciones se muestran en la Tabla 3-4.

Tabla 3-4: Recomendaciones de la serie Y de la ITU

Recomendación	Descripción
Y.2012	Requisitos y arquitectura funcional de una NGN.
Y.2013	Requisitos funcionales y arquitectura del marco de servicios convergentes.
Y.2000	Ejemplos de servicios suplementarios en la convergencia fijo-móvil.

**ETSI-TISPAN:** el Instituto Europeo de Estándares de Telecomunicaciones (ETSI, European Telecommunications Standards Institute) [50], es un organismo de estandarización cuyo principal logro es el aporte de diferentes estándares para redes GSM.

Cuando se dio el auge de la convergencia de las redes de telecomunicaciones, en el año de 2003 la ETSI creó el grupo TISPAN (Servicios convergentes de internet y telecomunicación & protocolos para redes avanzadas), dada la necesidad de integrar los estándares creados de la 3GPP para redes GSM, a las redes fijas. En el año 2005 el público conoció la versión 1 de la especificación para redes de nueva generación y en el año de 2008, la versión 2, que incluía IMS y capacidades IPTV.

**ONEAPI:** esta iniciativa define un conjunto de API's e interfaces Web, que permiten a los operadores de redes móviles y otros, exponer la información de la red y capacidades para desarrolladores [51]. Su objetivo es reducir el esfuerzo y el tiempo necesario para crear aplicaciones y contenido. Este conjunto de APIs son expuestas a través de diferentes técnicas software como REST y formatos como JSON, XML, entre otros.

OneAPI define una serie de casos de uso como:

- Servicios de Carga: permite a una tercera parte hacer cobros a un usuario a través del operador.
- Mensajería: permite a una aplicación web enviar y recibir SMS.
- Localización: permite a una aplicación web pedir la localización de uno o más dispositivos móviles que están conectados a un operador de red.
- Perfil de Usuario: permite determinar el estado de usuario y sus preferencias.
- Perfil de conexión de datos: Permite a una aplicación solicitar el tipo de canal portador que utiliza un usuario (3G, GPRS, etc.).

**TelcoML:** Es una iniciativa que nació desde el 2008, la cual busca representar servicios de telecomunicaciones y servicios IT, a través de UML. En el transcurso de estos años se hicieron grandes avances para que esta nueva iniciativa se convirtiera en estándar, lo cual se logró en 2011<sup>9</sup>.

En [52] se presentan las dos principales partes de TelcoML, las cuales son una librería habilitadora y un perfil de composición. El primero de estos representa en UML las principales API Telco; y el segundo, un modelo, también en UML, de los servicios compuestos ejecutables.

Los principales servicios que soporta esta especificación son: SMS & MMS, Click to Call, localización, sincronización, reconocimiento de voz y TTS.

### 3.2.2 Plataformas para el despliegue de servicios

**Mobicents:** el entorno de ejecución (JAIN SLEE) del grupo Mobicents cuenta con una serie de servicios por defecto disponibles al público por medio de documentos de descripción y de su código fuente para fines de implementación.

Algunos de los servicios ofrecidos por este entorno de ejecución son [32]:

---

<sup>9</sup> La especificación Beta de TelcoML se encuentra en <http://www.omg.org/spec/TelcoML/1.0/Beta1/PDF>

- Control de llamada: este servicio actúa como un centro de llamadas simple, con funcionalidades como correo de voz, desvío de llamadas y funciones de bloqueo.
- Compras: este servicio consta de una tienda en línea capaz de realizar llamadas a VOIP para informar a los usuarios o administradores acerca del estado de sus cuentas. Los usuarios pueden efectuar las decisiones de compra (aceptar, rechazar, establecer fecha de envío).
- Despertador SIP: esta aplicación procesa mensajes SIP de un Agente SIP registrado para actuar como un sistema despertador. Se utiliza un formato de mensaje específico el cual contiene un temporizador T, un mensaje despertador M y el usuario de destino. Cuando T se cumple el mensaje es enviado.

**Rhino Slee:** Rhino de Open Cloud [31] [53] ofrece una gran cantidad de servicios los cuales pueden ser implementados con su entorno de ejecución<sup>10</sup>. Algunos de estos se enumeran a continuación.

- Servicios de valor agregado a la llamada básica: terminación de llamada, distribución de llamada, límite a la duración de llamada, desvío de llamada, llamada en espera, identificador de llamada entrante, transferencia de llamada
- Servicios de multimedia: servidor de chat, publicidad durante la llamada, mensajería multimedia, video conferencia, centro de mensajería virtual.
- Servicios de mensajería: texto interactivo, retardo en la entrega de mensajes, traductor, notificación de llamada perdida, texto en tiempo real, información turística.
- Pagos y facturación: llamada patrocinada, supervisión de límite de crédito, notificación fondos insuficientes, tarjeta de llamadas virtuales, control de privacidad.
- Servicios Web: comunicaciones sociales, video conferencia Web, "Click to Call".

**Ericsson SDP:** esta SDP ofrece todas las fases de aprovisionamiento de servicios (creación, despliegue, monitoreo, facturación, retiro, etc.) [29]. Los principales servicios que integran esta plataforma son:

- Un conjunto de habilitadores de servicio utilizados por la capa de ejecución como: SMS, MMS, ubicación, presencia, mensajería instantánea.
- Funciones comunes del aprovisionamiento de servicios como: manejo de dispositivos, facturación, manejo de identidad.
- Acceso a datos: provee un perfil de usuario con sus servicios y características.

---

<sup>10</sup> La página de Desarrolladores de Rhino Slee, presenta alrededor de 35 ejemplos para la implementación de servicios.

- Acceso a contenido de terceros como servidores de medios (descarga de archivos, música, medios sociales, etc.).

### 3.2.3 Operadores de telecomunicaciones

Son abundantes los operadores de telecomunicaciones alrededor del mundo que prestan una gran variedad de servicios a sus usuarios dependiendo de la ubicación geográfica en la que se encuentran, algunos de ellos son nombrados a continuación.

**Vodafone:** Es un operador de carácter multinacional de telefonía móvil y telefonía fija. Ofrece servicios convergentes a sus usuarios entre los cuales se destacan:

- Contestador: permite al usuario convertir los mensajes que se encuentran en el buzón de voz a texto. Este servicio cobra importancia si el usuario se encuentra ocupado o en una reunión y no puede escuchar los mensajes que se encuentran en su buzón.
- Dicta SMS: permite al usuario que realiza una llamada dejar un mensaje de voz y si este lo desea, el sistema lo convierte a texto, para que el destinatario lo reciba en forma de SMS.
- Desvío de llamadas: permite al suscriptor desviar las llamadas que llegan a su fijo o móvil, a otros destinos; por ejemplo: si el usuario está atendiendo una llamada importante y su teléfono móvil se está quedando sin batería puede desviar las llamadas a otro número fijo o móvil para continuar hablando.
- Rellamada: cuando un usuario desea ponerse en contacto con otro y éste no se encuentra disponible, el servicio de rellamada permite realizar nuevamente una llamada de forma automática, cuando la red detecta que el usuario se encuentra disponible.
- Restricción de llamadas: permite restringir llamadas salientes, entrantes o internacionales.

**Tigo Colombia:** Es un operador de telecomunicaciones que presta servicios de telefonía móvil, algunos de los servicios que ofrece son [54]:

- Recarga: permite realizar recargas a través de un sitio web, mediante tarjeta de crédito o débito.
- Envío de mensajes: permite establecer conversaciones desde un teléfono móvil y un usuario que cuente con una conexión a internet.
- Tono de llamada (backtones): Permite escuchar una melodía predefinida mientras el usuario espera a que el destinatario conteste; puede ser configurado desde un sitio Web.
- Transferencia de saldo: permite a un usuario enviar saldo a un número Tigo.

- Entretenimiento: el usuario puede descargar diferentes tipos de contenido como: videos juegos, música, televisión, entre otros, a su teléfono móvil.

### **Claro Colombia:**

Ésta empresa surge de la alianza entre Comcel y Telmex Colombia el 26 de Junio de 2012. A partir de la fecha, los servicios que prestaba el operador móvil y el operador de Telecomunicaciones, son ofrecidos por una misma y unificada firma.

Los servicios ofrecidos son: Llamada, llamada en espera, llamada a elegidos, SMS, Video Conferencia, Localización y en general, todo paquete de servicios móviles antes ofrecidos por Comcel. Por otra parte, el conjunto de servicios ofrecidos por Telmex como: televisión Digital, Llamada SIP, Internet, configuración de correo electrónico, pagos, etc.

## **3.3 Modelo en capas de un entorno Telco 2.0**

Como se explicó en la Sección 2.1.1, un entorno Telco 2.0 permite la integración de servicios de Telecomunicaciones y Web, ya que ofrece las condiciones necesarias para la ejecución de servicios convergentes mediante la implementación de especificaciones como JAIN SLEE e IMS.

En la Figura 3-5 se ilustra el modelo en capas de un entorno Telco 2.0, la cual se basa en el trabajo de Al-Begain *et al.* [23]. Es importante resaltar que la capa de servicios se propuso a partir de la revisión bibliográfica que se realizó de servicios Web, Telco, convergentes y a organizaciones de estandarización (Sección 3.1 y 3.2). A continuación se describen las capas que la componen.

**Capa de aplicación:** comprende las plataformas que ejecutan la lógica de los servicios, que se encuentran en la *capa de integración* o directamente en la *capa de servicios*. Dichas plataformas pueden estar alojadas en el núcleo del operador, o en servidores remotos (ej: JEE, .NET).

**Capa de integración:** esta capa busca acelerar el despliegue de los servicios y el tiempo de salida al mercado, algunas plataformas encargadas de la lógica de los servicios utilizan tecnologías IT para sistemas distribuidos como las que siguen los principios de SOA. La capa de integración facilita la conexión entre los contenedores de servicios con servicios reutilizables compuestos por servicios web o Telco lo cual permite la creación de servicios convergentes. Adicionalmente se ofrece funcionalidades para la gestión de la arquitectura de servicios así como políticas y mecanismos de seguridad para los servicios en tiempo de ejecución.

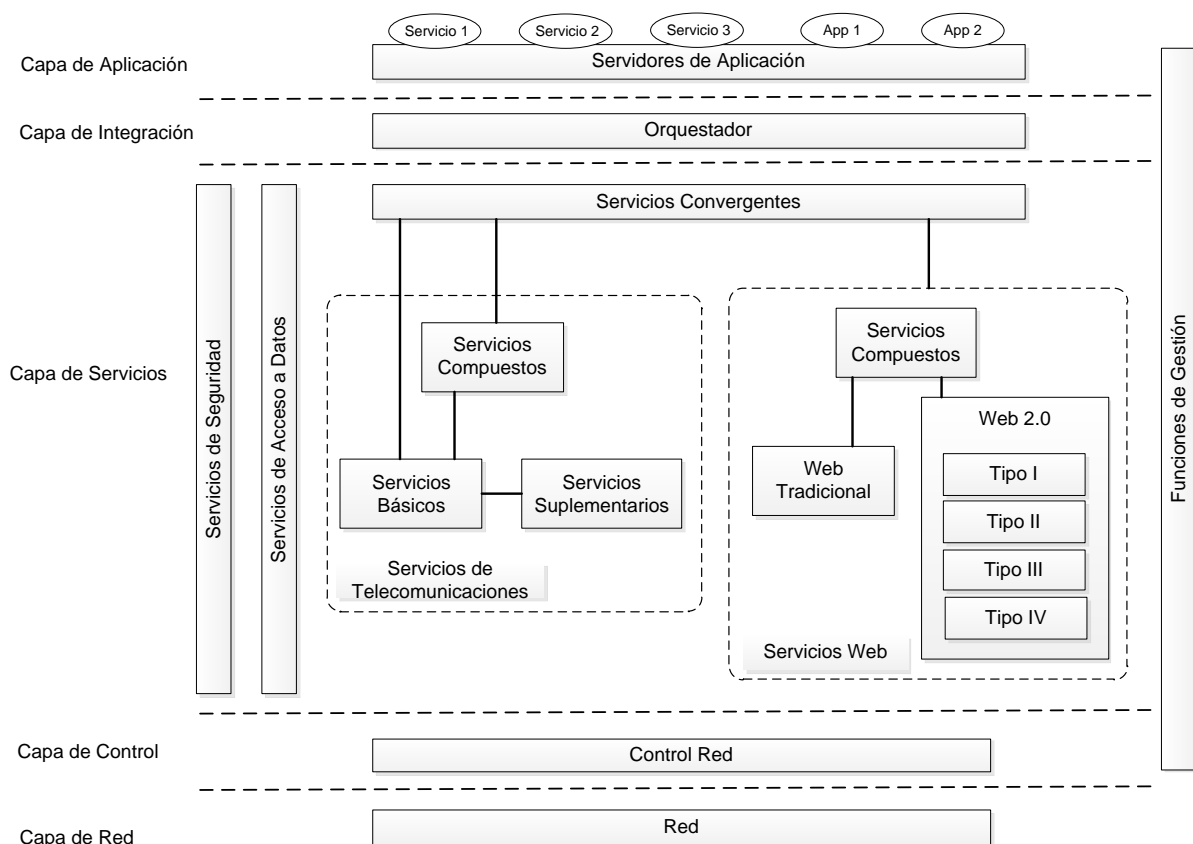


Figura 3-5: Capa de servicios propuesta teniendo en cuenta la arquitectura en capas de un entorno Telco 2.0 de [23]

**Capa de servicios:** aloja un conjunto de servicios los cuales están caracterizados por servicios de Telecomunicaciones, Web, Convergentes, de acceso a datos y seguridad. Estos pueden ser accedidos mediante diferentes protocolos y APIs; y adicionalmente pueden ser reutilizados para la composición de nuevos servicios. Comprende todos los servicios establecidos en el Catálogo de Servicios de la Sección 3.4.

**Capa de control:** ésta capa reúne todos los elementos para implementar el plano de control el cual representa la inteligencia de los servicios básicos en la red. En entornos Telco 2.0 esta capa puede ser implementada por JAIN SLEE, haciendo uso de adaptadores de recurso; y por la arquitectura IMS.

**Capa de red:** proporciona el acceso y transporte necesario para el tráfico de cada servicio. La capa de Red está constituida por la red de acceso y la red núcleo donde se encuentran los servicios portadores y permite a un usuario acceder al dominio de prestación de servicios a través de un terminal.

Una vez ilustrado el modelo en capas (Figura 3-5), se puede observar el conjunto de servicios que hacen parte de un entorno Telco 2.0 (capa de servicios), sin embargo, para la elaboración del catálogo de servicios (objetivo uno, Sección 3.4) y la detección de patrones de flujo de control y datos (objetivo dos, Capítulo 4) únicamente se tuvo en cuenta la capa de

servicios, donde se encuentran servicios de Telecomunicaciones, Web y Convergentes. Por lo anterior, la capa de red y el resto de capas no serán objeto de estudio.

## 3.4 Catálogo de servicios Propuesto

Teniendo en cuenta la revisión realizada en la Sección 3.2, a continuación se describe el catálogo de servicios propuesto. Cada servicio es clasificado según la arquitectura de servicios propuesta, la cual se ilustra en la Figura 3-5 (Capa de Servicios).

### 3.4.1 Servicios de Telecomunicaciones

#### Servicios Básicos

El catálogo de servicios de Telecomunicaciones Básicos se muestra en la Tabla 3-5.

Tabla 3-5: Catálogo de Servicios: Servicios de Telecomunicaciones - Básicos

Nombre	Descripción
Llamada	Comunicación mediante transmisión de voz.
SMS	Servicio de envío de mensajes.
Video Llamada	Comunicación mediante transmisión de voz y video.
Transferencia de Datos	Permite la conexión en tiempo real entre dos puntos extremos con el fin de transferir diferentes tipos de contenidos.

#### Servicios Suplementarios

El catálogo de servicios de Telecomunicaciones Suplementarios se muestra en la Tabla 3-6.

Tabla 3-6: Catálogo de Servicios: Servicios de Telecomunicaciones - Suplementarios

Nombre	Descripción
Identificación de llamada	Es un servicio que permite conocer el número de la persona que realiza una llamada.
Numero privado	Es un servicio que permite ocultar el número del cual se realiza la llamada.
Desvío de llamada	Permite a un usuario realizar la configuración de redirigir las llamadas entrantes hacia otro número previamente definido
Presentación de los datos de usuario	Es un servicio donde se muestra la identificación (establecido en 80 caracteres), del usuario de la red.
Llamada en espera	Este servicio se divide en: Llamada en espera: cuando un usuario A se encuentra en comunicación con el usuario B y un usuario llama a A, este puede colocar en llamada en espera al usuario B, para atender la llamada de C. Llamada retenida: permite interrumpir una llamada y si lo desea, volver a activarla o terminar definitivamente con esta
Llamadas múltiples	Servicio que permite establecer múltiples llamadas simultáneamente.
Aviso de carga	Información del costo de la llamada para el usuario de la red móvil.
Señalización usuario a usuario	Es un servicio que permite transmitir información del usuario, al momento de iniciar, terminar o durante la llamada



Bloqueo de todas las llamadas entrantes/salientes	Este servicio hace posible que un suscriptor tenga la capacidad de restringir las comunicaciones salientes, de acuerdo a un programa de restricción de llamadas.
Transferencia de llamada explícita	Este servicio adiciona capacidades al servicio de llamada en espera. Se tienen en cuenta tres usuarios A, B y C. El usuario A recibe o realiza las llamadas a B y C, donde A tiene la capacidad de invocar este servicio, para establecer la comunicación entre los Usuarios B y C.
Prioridad de llamada	Servicio que permite establecer el nivel de prioridad de una llamada, según el usuario lo desee. El nivel de prioridad más alto (A), es utilizado por ejemplo, para llamadas de emergencia, el segundo nivel (B) puede ser reservado para la red, y adicionalmente se presentan 5 niveles de prioridad (0-4).

### Servicios Compuestos de Telecomunicaciones

El catálogo de servicios de Telecomunicaciones Compuestos se muestra en la Tabla 3-7.

Tabla 3-7: Catálogo de Servicios: Servicios de Telecomunicaciones - Compuestos

Nombre	Descripción
Servicios de mensajería instantánea	Se define como un tipo de servicio de comunicaciones basado en texto que permite al usuario crear una especie de conversatorio (chat room) con otro individuo en tiempo real
Pulsar Para Hablar (PTT, Push To Talk)	Servicios orientados a desplegar el servicio PTT usando elementos de red de la NGN, con el fin de permitir a una persona establecer una sesión multimedia con un grupo en particular pulsando sólo un botón Este servicio se ofrece a través de una red de telefonía celular y es llamado servicio de pulsar para hablar sobre el celular o PoC (Push To Talk over Cellular)
Conversión de voz a texto:	Es un servicio que convierte los mensajes de voz a texto. Una vez convertido, se envía el contenido al destinatario a través de un SMS.
Escucha Directa	Es un número corto que se incluye en el SMS y permite oír directamente el mensaje de voz que un usuario acaba dejar. Desde el teléfono móvil se puede realizar una llamada a este número para escuchar el mensaje.
Dicta SMS:	Para quien llama: cuando un usuario llama a un teléfono pero el destinatario no contesta o rechaza la llamada, se escuchará una locución gratuita que ofrecerá la posibilidad de grabar un mensaje para el número receptor. El usuario espera hasta oír la señal (bip) y deja un mensaje de voz que llegará como SMS al destinatario.  Para quien tiene el "Dicta SMS" activado: cuando un usuario no puede responder o rechaza una llamada, la persona que llama podrá grabar un mensaje que será recibido en formato SMS.
Desvío de Llamadas:	Con este servicio es posible desviar las llamadas que lleguen a un móvil para que pueda recibirlas en un fijo o en otro móvil, o para que vayan directamente a un contestador.

Restricción de llamada por el usuario o el operador	Con este servicio es posible restringir la recepción o emisión de determinadas llamadas, tanto las que un usuario realiza, como las que recibe. Es útil si se quiere limitar el uso del teléfono móvil de forma genérica o por diferentes tipos de llamadas.
Servicios personalizados por el usuario	Servicios de telecomunicaciones creados por un usuario determinado por medio de la unión de servicios básicos y servicios suplementarios

### 3.4.2 Servicios Web

#### Servicios Web Tradicionales

El catálogo de servicios Web Tradicionales se muestra en la Tabla 3-8.

Tabla 3-8: Catálogo de Servicios: Servicios Web - Tradicionales

Nombre	Descripción
Pagos	Servicio de pagos. Se puede realizar antes de obtener el servicio o después de consumirlo.
Llamada a un tercero	Creación y manejo de una llamada iniciada por una aplicación
Notificación de llamada	Este servicio proporciona funciones simples para desarrolladores de aplicaciones para determinar cómo debe ser tratada una llamada, por ejemplo, si es necesario finalizar, continuar, o redirigir una llamada.
Manejo de cuenta	Consulta de Cuenta, Recarga Directa y Recarga a través de comprobantes o "vouchers".
Estado del terminal	Obtener el estado de un terminal (Alcanzable, Inalcanzable u Ocupado).
Localización de terminal	Obtener la información sobre la ubicación de un Terminal.
Manejo de lista de direcciones	Manejo de grupos (alias) de suscriptores.
Presencia	Información de presencia (en línea, ocupado, no disponible).
Geocoding	Obtener la dirección de ubicación de un suscriptor, EJ: país, estado, distrito, ciudad, calle, número de casa, información adicional, código postal.
Capacidades del dispositivo y configuración	Obtener información relacionada con las capacidades de un dispositivo y enviar la configuración de este a otro.
Multimedia streaming control	Control de streaming (flujo de datos) de multimedia a un suscriptor. EJ: Transferencia entre terminales de usuarios.
Manejo de contenido	Permite subir y consumir contenido de la red (o de un proveedor de contenido).
Políticas	Ofrece aprovisionamiento y las funciones de evaluación para políticas.
Consulta de información	Acceso a un sitio Web para obtener información, por ejemplo: búsqueda en diccionarios, consulta de saldo, recarga de celular, compra boletos, traducción de textos, etc.

#### Servicios Web 2.0

Retomando el modelo de creación de conocimiento (Sección 3.1.2). el catálogo de servicios Web 2.0 se muestra en la Tabla 3-9.

Tabla 3-9: Catálogo de Servicios: Servicios Web - Web 2.0

<b>Tipo</b>	<b>Nombre</b>	<b>Descripción</b>
<b>Intercambiador (Tipo I)</b>	MSN	Abreviación de MicroSoft Network, conjunto de servicios que incluyen mensajería instantánea, correo electrónico, Volp, etc.
	Skype	Software que permite la comunicación mediante texto, voz y video sobre internet (Volp), empleando el modelo punto a punto (P2P).
	Google Talk	Es un cliente de mensajería instantánea y VoIP que hace uso del protocolo extensible de mensajería y comunicación de presencia (XMPP).
<b>Agregador (Tipo II)</b>	MySpace	Sitio web donde los usuarios comparten sus pensamientos, fotos, actividades, entre otros. Cuenta con servicios de noticias, clasificados, videos, aplicaciones para móviles, etc, además de apis para desarrolladores.
	Youtube	Es un sitio web donde los usuarios pueden subir y compartir videos. Utiliza la tecnología Flash para reproducir los videos en línea.
	Facebook	Sitio web de red social, inicialmente creado para que estudiantes de la universidad de Harvard se mantuvieran conectados, pero en la actualidad cualquier usuario con una cuenta de correo electrónico puede acceder y compartir sus pensamientos, fotos, videos y desde el año 2011 la posibilidad de realizar videollamadas.
	Twitter	Es una red social que permite compartir a sus usuarios información mediante: texto de máximo 140 caracteres y fotos; cuenta con una api abierta para desarrolladores.
	43things	Es una red social basada en los principios del etiquetado (tagging). Los usuarios ingresan una serie de gustos y expectativas, y dependiendo de estas, los usuarios son conectados a otros, que compartan características similares.
	Linkedin	Es una red social orientada a negocios y a profesionales. Este sitio permite a los usuarios, encontrar ofertas laborales, realizar negocios, conocer gente que comparte sus mismos gustos profesionales, entre otros.
<b>Colaborador (Tipo III)</b>	Answers.com	Es un sitio basado en el intercambio de conocimiento sobre internet. Permite a los usuarios escribir preguntas o consultar preguntas previamente respondidas sobre diferentes temas.
	Saleforces.com	Es un sitio enfocado a desarrolladores donde estos pueden desarrollar sus aplicaciones, programas, etc. en línea, gracias a una plataforma que está en un servidor.
	Yahoo Widget	Es un sitio web que permite crear widgets en línea basados en la tecnología de JavaScript, AdobeFlash entre otros.
	Wikipedia	Es un proyecto para escribir comunitariamente una enciclopedia libre. Los usuarios pueden editar el contenido de los artículos de manera muy simple.
<b>Liberador (Tipo IV)</b>	Openoffice	Es una plataforma ofimática <sup>11</sup> de libre distribución que incluye: Procesador de textos, hojas de cálculo, presentaciones, etc. Ésta disponible en varias plataformas

<sup>11</sup> Ofimática: es una recopilación de aplicaciones, las cuales son utilizados en oficinas y sirve para diferentes funciones como: crear, modificar, organizar, escanear, imprimir.

		como: Microsoft Windows, GNU/Linux, BSD, Solaris y Mac OS.
	Linux	Linux es el Núcleo del sistema operativo GNU/Linux. Es el ejemplo más representativo de software libre; todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General).

### Servicios Web Compuestos

El catálogo de servicios Web Compuestos se muestra en la Tabla 3-10.

Tabla 3-10: Catálogo de Servicios: Servicios Web -Compuestos

Nombre	Descripción
Planeación de viajes:	Permite al usuario: reservar su vuelo, el hotel donde se desea hospedar, los recorridos por la ciudad, el alquiler de vehículos, etc.
Compras	El usuario escoge en una aplicación web una variedad de artículos y los agrega a un carrito de compras, realiza el pago con tarjeta de crédito y adicionalmente se hace una llamada a un número telefónico mediante VoIP, para reconfirmar su compra.
Ubicación de mercancía	Ubica en qué lugar se encuentra la mercancía de un cliente y permite saber si esta ya fue entregada o cuantos días faltan para su entrega.
Servicios Bancarios	Notificación de transferencias, retiros, consignaciones, extractos.
Servicios de comunicación interactiva colaborativa:	Todo el conjunto de servicios para el soporte de conferencia multimedia con disponibilidad de compartir archivos y aplicaciones, como e-learning y juegos.
Servicios de despliegue de contenido:	Aplicaciones para la entrega de video y otros flujos de medios a los usuarios. Entre ellos se encuentran la radio, música, difusión de video, video bajo demanda, distribución de canal de televisión digital, distribución de información financiera, y distribución de contenido médico y profesional
Servicios de información	Por ejemplo, servicios de disponibilidad de boletos, de estado de tráfico, servicios avanzados de notificación (push), etc.
Servicios basados en localización	Entre ellos, servicios de guía turística, servicios de asistencia médica y hospitalaria, etc.
Servicios personalizados por el usuario	Servicios web creados por un usuario determinado por medio de la unión de servicios de la web tradicional o web 2.0 y cualquier combinación compatible entre estos dos.

### 3.4.3 Servicios Convergentes

El catálogo de servicios Convergentes se muestra en la Tabla 3-11.

Tabla 3-11: Catálogo de Servicios: Servicios Convergentes

Nombre	Descripción
Servicio convergente de voz y presencia	Este servicio notifica al servicio presencia que el dispositivo de usuario está registrado y disponible para mensajería instantánea, de esta forma los usuarios en la lista de contactos son notificados que el usuario está disponible. Cuando el usuario recibe una llamada el servicio de presencia cambia su estado a “no molestar durante la llamada” el cual puede ser observado por todos los contactos.
Servicio convergente reenvío de llamada y videoconferencia	Cuando el usuario inicia el servicio de videoconferencia, su terminal cambia de estado y pasa a ocupado, con lo cual la red cambia su comportamiento en respuesta a futuras llamadas. Esta vez, si entran otras llamadas, deben ser reenviadas automáticamente a otro terminal
Movimiento de llamada conferencia de un cliente fijo a un cliente móvil	Servicio en el cual un usuario "A" mantiene una videoconferencia con "B". "A" decide cambiar de ubicación pero no desea interrumpir su conversación por el dispositivo móvil. Para ello, puede apretar un botón en su teléfono móvil y la parte audio de la llamada pasará automáticamente a su teléfono móvil.
Compras	Este servicio consiste de una tienda online capaz de realizar llamadas VOIP para informar a los usuarios o administradores a cerca del estado de sus cuentas. Los usuarios llamados puede afectar las decisiones de compra (aceptar, negar, establecer fecha de envío).
Facebook Event Reminder	Es un servicio que utiliza los servicios Telco de llamada y SMS, para recordarle a un usuario que se encuentra registrado en el servicio, los eventos de Facebook a los que este va a asistir que estén próximos a iniciar.
Confirmación de compras en Ebay	Es un servicio que complementa las funcionalidades del servicio Web 2.0 de Ebay. Al momento de finalizar el proceso de compra, el usuario recibe una llamada de confirmación, con el fin de evitar fraudes por suplantación de identidad.
CallMessage	Realiza una llamada a un usuario; en caso de que este no conteste, el servicio envía automáticamente un mensaje directo a Twitter, la información del usuario que lo llamo, la fecha y la hora.
TwitterSMS	El usuario ingresa a una aplicación Web, la cual ofrece dos servicios: enviar un SMS o enviar un mensaje directo a Twitter.
TwitterCalls	Servicio que permite llamar, enviar SMS, o dar un tweet a las personas que “te siguen”, o las personas que “sigues”, con la condición de estar registradas en un portal Web.

#### 3.4.4 Acceso a datos

Existen algunos estándares que definen la forma el acceso a datos de una aplicación o servicio, uno de ellos es JDBC, con el cual las aplicaciones desarrolladas en el lenguaje java pueden acceder de forma uniforme a las bases de datos independientemente del SGBD que se esté utilizando. De esta manera, este tipo de aplicaciones no necesitan conocer las especificaciones de un SGBD en particular, basta con comprender el funcionamiento de JDBC

Se han definido algunos métodos de desarrollo que pueden hacer uso de JDBC como driver para acceso a datos, algunos de estos métodos software son:

- DAO: un Data Access Object (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.
- JPA: Java Persistence API, es la API para acceso a datos desarrollada para la plataforma Java EE.

Persistencia en este contexto cubre tres áreas:

- La API en sí misma, definida en `javax.persistence.package`.
- La Java Persistence Query Language (JPQL).
- Metadatos objeto/relacional.

El entorno de ejecución JAIN SLEE de Mobicents utiliza un adaptador de recursos JDBC el cual le permite adaptar fuentes de datos (datasources) JDBC con el dominio JAIN SLEE y proporciona los medios necesarios para ejecutar declaraciones en forma asíncrona. De esta forma este adaptador de recursos envía resultados a través de eventos liberando al enrutador de eventos de la carga producida por los recursos utilizados en las interacciones hechas con las bases de datos, de igual forma las aplicaciones JAIN SLEE son completamente libres de tener que gestionar el cierre de la conexión [55].

Estas herramientas ofrecen grandes ventajas para aplicaciones o servicios en el momento de gestionar cierta información, por ejemplo el acceso a través de Internet posibilita una forma rápida y cómoda de acceder a la información almacenada en bases de datos con el fin de recuperarla, modificarla o insertarla. Se despliega así todo un conjunto de posibilidades para el diseño de páginas Web, servicios Telco o Web, aplicaciones, etc., que ofrecen servicios como los siguientes [56].

- Catálogos en línea, con información siempre actualizada sobre productos, información sobre visitantes, que puede actualizarse dinámicamente a medida que éstos navegan por el sitio web.
- Generación en tiempo real de páginas, en función de selecciones o gustos del cliente, o del momento del día, o cualquier otro criterio.
- Conservación del contenido de un sitio web en una BD, que se cargue dinámicamente cuando el usuario solicita una página, de forma que separe el diseño y formato de las páginas de su contenido.
- Servicios en línea como correo web, acceso remoto a BD corporativas, noticias, guías de servicios.

### 3.4.5 Seguridad

Existen protocolos encargados de prestar seguridad para el intercambio de un terminal y otro. Estos actúan en diferentes capas de modelo OSI por ejemplo:

- Protocolo de seguridad en internet (IPsec, Internet Protocol security): actúa en la capa 3 (red), cifrando cada paquete IP en un flujo de datos. Su uso es de carácter obligatorio en IPv6 y puede ser implementado en IPv4 [57].
- Capa de conexión segura (SSL, Secure Sockets Layer): es un sistema que proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente, con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA. Este sistema puede funcionar en conjunto con el protocolo HTTP, SMTP, FTP en la capa de aplicación del modelo TCP/IP [58].
- Protocolo seguro de transferencia de hipertexto (S-HTTP, Secure HyperText Transfer Protocol): S-HTTP está integrado con HTTP. Aquí, los servicios de seguridad se negocian a través de las cabeceras y atributos de la página. Por lo tanto, los servicios de S-HTTP están disponibles sólo para las conexiones de HTTP [59].

### 3.5 Resumen

Este Capítulo presentó una síntesis que enmarca a los servicios Web y de Telecomunicaciones, mediante la revisión bibliográfica de conceptos, definiciones, organizaciones de estandarización, plataformas para el despliegue de servicios y operadores de telecomunicaciones. Seguido a esto, se propuso un diagrama en módulos el cual permite clasificar servicios Telco, Web y Convergentes en un entorno Telco 2.0; y finalmente se realizó un catálogo de servicios.





## Capítulo 4

### 4 Detección de patrones de flujo de ejecución en servicios Telco 2.0

En este Capítulo, se realiza la detección de patrones de flujo de ejecución presentes en un conjunto de servicios Telco 2.0, definidos en el catálogo (sección 3.4).

La Figura 4-1 muestra un esquema de cuatro módulos, los cuales describen el proceso de detección de patrones de flujo de ejecución. El primer módulo describe los patrones de flujo de control (CFP) y datos (DP) (Sección 4.1), los cuales son estudiados y evaluados en [7], [60] y sirven como referencia en el proceso de detección realizado en este trabajo. En el segundo módulo se presenta el modelado y descripción, mediante una representación formal de un conjunto de servicios Telco 2.0, sobre la cual se va a detectar patrones de flujo de ejecución (sección 4.2). El tercer módulo explica el método para la detección de CFP y DP donde se hace uso de un algoritmo de detección de subestructuras para los CFP y se analiza el soporte de JAIN SLEE en el flujo de datos de los servicios (Sección 4.3). Finalmente, el cuarto módulo corresponde a la lista de los patrones detectados (Sección 4.4).

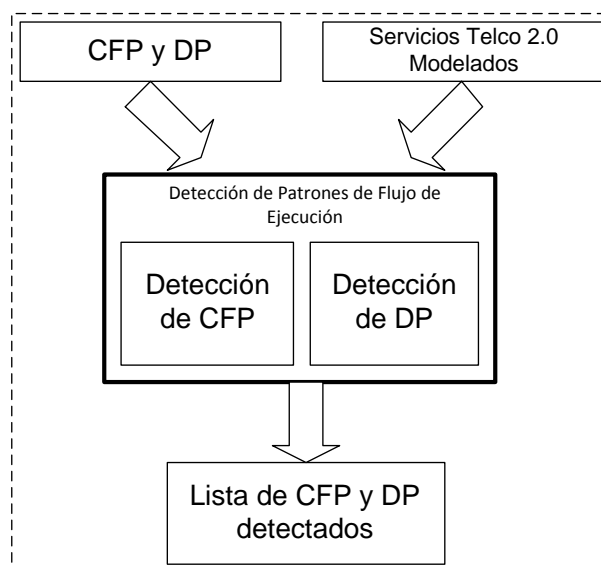


Figura 4-1: Esquema en módulos para la detección de patrones de flujo de ejecución en servicios Telco 2.0

El proceso de detección se realiza con el fin de identificar los patrones de flujo de ejecución que soportan la composición de servicios convergentes, los cuales son la base para la construcción del prototipo experimental (Capítulo 5).

Adicionalmente, es importante resaltar que el análisis de los patrones de flujo de control y datos, no se limitó a servicios Telco 2.0 (convergentes), ya que también se tuvo en cuenta los patrones presentes en servicios de telecomunicaciones (básicos y compuestos) y Web (tradicionales y Web 2.0).

## 4.1 Análisis de patrones de Flujo de Ejecución

A continuación se presenta la descripción de los patrones de flujo de control (Sección 4.1.1) y datos (Sección 4.1.2).

### 4.1.1 Patrones de flujo de control

El estudio de CFP realizado por Vander Aalst *et al.* [7] [30] identificó un conjunto de 43 estructuras recurrentes independientes de su implementación. Este trabajo realizó evaluaciones a diferentes sistemas de Workflow como: WebSphere BPEL, Oracle BPEL, BPMN, entre otros, donde se identificó cuáles de estos eran capaces de implementar los CFP.

Los patrones propuestos por Vander Aalst se clasifican en las siguientes subcategorías:

- **Patrones básicos:** esta clase de patrones captura los aspectos elementales del flujo de control (ej. patrón de secuencia, patrón de división paralela).
- **Patrones de ramificación y sincronización avanzada:** es una serie de patrones que se caracteriza por utilizar conceptos más complejos de ramificación y sincronización (ej. patrón de combinación simple, patrón de múltiple escogencia).
- **Patrones de múltiples instancias:** describen las situaciones en donde hay múltiples hilos de ejecución activos en un modelo de proceso (ej. patrón de múltiple instancia sin sincronización).
- **Patrones basados en estados:** estos muestran la situación donde las soluciones son más fácilmente ejecutadas en los lenguajes de procesos que soporten la noción del estado. En este contexto se consideran los estados de una instancia de un proceso, para incluir una amplia colección de datos asociados con la ejecución actual incluyendo el estado de varias actividades, así como también la información relevante al proceso, tal como los elementos de las actividades y los elementos de datos (ej. patrón de selección diferida).
- **Patrones de cancelación y de terminación forzada:** existen patrones que tienen variantes que utilizan el concepto de cancelación de actividades, donde las instancias de actividades son activadas o desactivadas (ej. patrón de terminación implícita).

- **Patrones de iteración:** capturan un comportamiento repetitivo (ej. patrón de ciclo estructurado).
- **Patrones de terminación:** estos abordan las circunstancias sobre las cuales un workflow se considera como completo (ej. patrón de cancelación de actividad).
- **Patrones de activación:** éstos se encargan de manejar las señales externas que pueden ser requeridas para iniciar ciertas actividades (ej. patrón de lanzador transitorio).

Cada una de las categorías nombradas anteriormente tiene asociado un determinado número de CFP [7] los cuales son descritos y representados mediante redes de Petri, ya que estas permiten capturar diferentes comportamientos de los patrones como sincronismo, paralelismo, concurrencia entre otros.

Como ejemplo, en la Figura 4-2 se muestra el *patrón de selección diferida* (Deferred Choice). Este patrón permite la divergencia del control de flujo de una rama en dos o más ramas tal que el flujo de control se direcciona a una actividad específica dependiendo del resultado de una actividad precedente. (Ver Anexo A. Patrones de Flujo de Control).

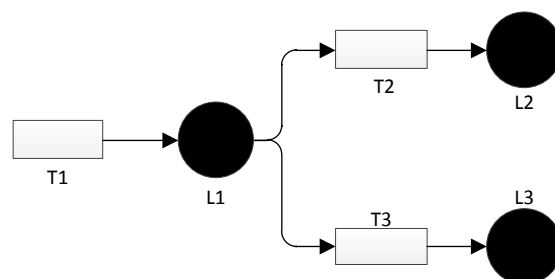


Figura 4-2: Patrón de selección diferida (Deferred Choice) representado mediante redes de Petri

#### 4.1.2 Patrones de Datos

En [8] se definen un conjunto de 40 DP, los cuales tienen como objetivo común, capturar las diversas formas en las cuales se representan los datos. Estos patrones son independientes del lenguaje de modelado y de la tecnología utilizada.

Para identificar DP, Van der Aalst *et al.* definen un modelo de workflow [61] el cual se muestra en la Figura 4-3, este modelo lo componen tareas (task) las cuales se conectan entre ellas en forma de grafos dirigidos. Adicionalmente, las instancias de un workflow son llamadas *casos* (case), y pueden existir múltiples instancias ejecutándose simultáneamente.

Dentro de un workflow se definen 4 tipos de tareas: atómicas (task), tareas en bloque (block task), tareas de múltiple instancia (multi-instance task) y tareas de múltiple instancia en bloque (multi-instance block task). El paso de datos que se da entre las tareas nombradas, datos provenientes de ambientes externos del workflow, y datos que provienen de tareas humanas, definen el conjunto de 40 DP, los cuales son explicados en detalle en [8].

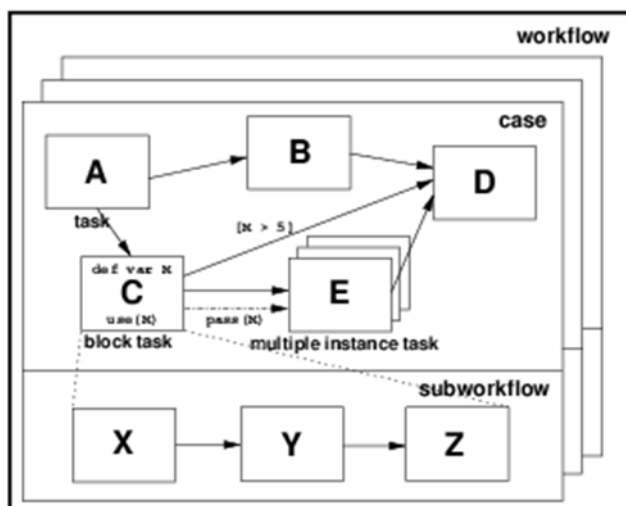


Figura 4-3: Modelo de Workflow para identificar DP

El conjunto de patrones es evaluado en algunos lenguajes de modelado comerciales como: Stafware Process Suite version 9, MQSeries Workow 3.3.2, XPDL 1.0 entre otros.

La Tabla 4-1 muestra los lenguajes evaluados, y los datos que estos leguajes soportan

Tabla 4-1: Correspondencia entre lenguajes de modelado y el tipo de datos que soportan

	Stafware	MQSeries	FLOWer	COSA	XPDL	BPEL4WS
String	+	+	+	+	+	+
Integer	+	+	+	+	+	+
Float	+	+	+	+	+	+
Boolean			+	+	+	+
Date	+		+	+	+	+
Time	+		+	+		+
document/reference	+		+	+	+	+
Enumeration					+	+
Composition	+	+	+		+	+
Array		+	+		+	
Set						+

Al igual que los CFP, Vander Aalst clasifica los DP en categorías las cuales se describen a continuación:

- **Visibilidad de Datos:** se refieren a la manera en que los elementos de datos pueden ser vistos por diversos componentes de un proceso de flujo de trabajo.
- **Interacción de Datos:** aquí se examinan las diversas formas en que los datos se transmiten entre los componentes en un proceso y cómo las características de los componentes individuales pueden influir en la forma en que el tráfico de elementos de datos se produce. De especial interés, es la distinción entre la comunicación de datos entre los componentes dentro de un proceso, como la interacción de datos orientada de un elemento de proceso con el ambiente externo.

- **Transferencia de Datos:** aquí se considera la manera en que la transferencia de datos se produce entre un componente de proceso y otro. Estos patrones sirven como una extensión a los DP de interacción y su objetivo es capturar a los diversos mecanismos por los cuales los elementos de datos se pueden pasar a través de la interfaz de un componente del proceso.
- **Enrutamiento de datos:** estos patrones muestran el comportamiento de la transmisión de datos entre tareas, las cuales son disparadas externamente. Tienen especial énfasis en la perspectiva de control.

Como ejemplo, en la Figura 4-4 se muestra el *patrón copia de datos*, el cual hace parte de la categoría transferencia de datos. Éste representa el paso de datos entre nodos, transiciones y una aplicación externa con características de almacenamiento de datos, por ejemplo, bases de datos y repositorios.

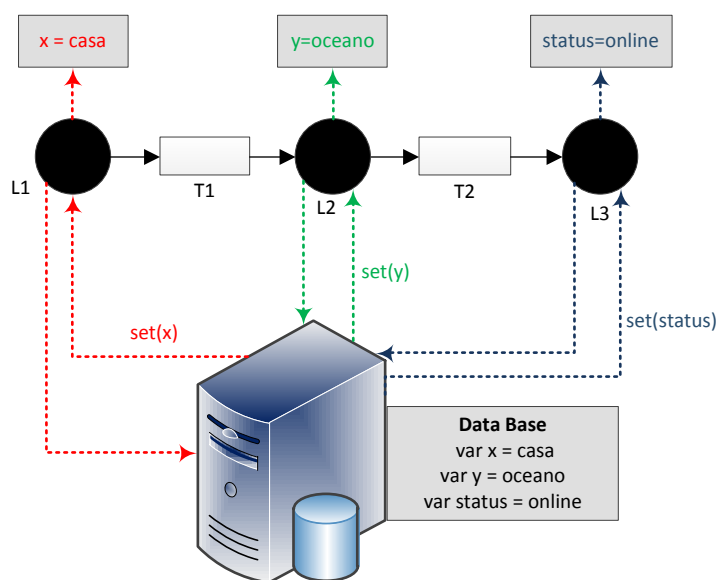


Figura 4-4: Patrón: copia de datos

## 4.2 Modelado de servicios Telco 2.0

En esta sección se divide en dos partes principales; en la Sección 4.2.1 se presenta una descripción general de la representación formal de redes de Petri, la cual permite identificar el flujo de control y datos de un servicio; posteriormente en la Sección 4.2.2 se explica el modelado de servicios Telco, Web y Convergentes utilizando dicha representación.

### 4.2.1 Redes de Petri

Una Red de Petri es un lenguaje formal y gráfico para modelar sistemas y procesos [62]. Estas se componen de lugares, transiciones, aristas o arcos y marcadas. Los lugares

representan los estados de la Red y estos a su vez pueden contener marcas. Las aristas o arcos de entrada conectan lugares con transiciones, mientras que las aristas o arcos de salida inician en una transición y terminan en un lugar.

Las Redes de Petri proporcionan una herramienta para describir sistemas caracterizados por ser concurrentes, asíncronos, distribuidos y no determinísticos. Desde un punto de vista gráfico, una Red de Petri puede ser utilizada como una ayuda de comunicación visual, de forma similar a las notaciones de diseño estructural del análisis de sistemas y las metodologías de diseño tradicionales. El lenguaje de las Redes de Petri, proporciona también una base matemática sólida para la descripción y análisis de ecuaciones de estado, modelos algebraicos y otros modelos matemáticos [63].

En la Figura 4-5 se puede observar un ejemplo de una red de Petri, la cual está compuesta por cuatro lugares (P0, P1, P2, P3 y P4) y cuatro transiciones (E0, E1, E2 y E3). El flujo comienza en P0, continua por E0, diverge en dos ramas (P1-E1-P3 y P2-E2-P4), converge en la transición E3 y finaliza en P0.

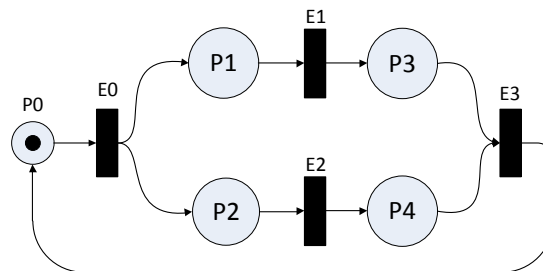


Figura 4-5: Ejemplo de una red de Petri

Este modelo permite representar el flujo de ejecución de los servicios estudiados, ya que capturan el flujo de control y el paso de datos entre lugares y transiciones; adicionalmente permite identificar los CFP presentes en dichos servicios [7] [8] los cuales usan la misma representación.

#### 4.2.2 Modelado de servicios en redes de Petri

Esta sección muestra el proceso que se siguió para modelar en redes de Petri, un conjunto de treinta servicios (Telco, Web y convergentes) del catálogo presentado en el Capítulo 3.

##### 4.2.2.1 Servicios Telco

El modelado de servicios Telco se realizó con base en las especificaciones técnicas de la ITU y la 3GPP, donde se describe el funcionamiento de los servicios básicos y suplementarios. Estas especificaciones definen características como: el número de usuarios que participan en un servicio, las capacidades ofrecidas a los usuarios, entre otras. Estas características son capturadas por la red de Petri a través de:

- Transiciones (redes de Petri<sup>12</sup>): representan que ha ocurrido un evento el cual ha sido generado por un usuario o por el mismo servicio dependiendo de su funcionamiento.
- Lugares (redes de Petri<sup>13</sup>) los cuales representan estados donde el usuario interactúa, o estados donde el servicio se encuentra esperando a que ocurra un evento.

Un ejemplo del modelado de servicios en redes de Petri es el servicio de *llamada básica*, el cual se ilustra en la Figura 4-6. Este servicio lo componen tres Lugares: “Ring”, “Talking” y “Busy Recording”; y cinco transiciones: “invite”, “busy”, “time out”, “bye” y “200 OK”. Para describir el funcionamiento del servicio se identifican tres usuarios: A, B y C. El servicio inicia en el estado Idle, cuando A realiza una invitación (Invite) al usuario B, en ese momento pasa al lugar “Ring”, donde puede ocurrir tres posibilidades: **I)** El usuario B se encuentra como ocupado y no puede contestar la llamada; **II)** Después de un tiempo determinado B no responde la invitación a hablar; y **III)** Los usuarios A y B se comunican satisfactoriamente. En **I**, se dispara la transición “Busy”, que activa el lugar “Busy Recording” donde el usuario A escucha una grabación que le indica la imposibilidad de contestar por parte de B, una vez finalizada la grabación se activa la transición “Bye” y el servicio regresa a su estado Inicial. En **II**, el servicio pasa a la transición “time out”, ya que ha transcurrido un tiempo determinado en el lugar “Ring” y la invitación a hablar no ha sido respondida, lo cual, finaliza el servicio. Finalmente, en **III**, la transición “200 OK”, indica que la invitación a hablar es contestada satisfactoriamente, activando así, el lugar “Talking”, donde A y B se comunican; en este lugar puede llegar una invitación a hablar por parte de C, ya sea a A o B, sin embargo, la llamada no puede ser contestada.

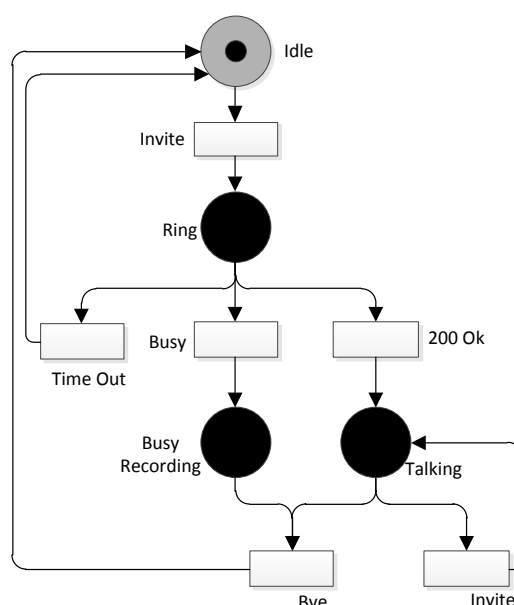


Figura 4-6: Servicio de llamada básica representado mediante a) redes de Petri.

<sup>12</sup> En redes de Petri cuando se dispara un evento se habla de transiciones.

<sup>13</sup> En redes de Petri, cuando el servicio se encuentra en un estado, por ejemplo: “hablando”, se habla de “lugares” o “Places”.

La descripción, funcionamiento y modelado del servicio anteriormente nombrado y de un conjunto de diez servicios Telco (básicos, suplementarios y compuestos), se encuentra en el Anexo C.

#### 4.2.2.2 Servicios Web

Para representar servicios Web a través de redes de Petri basta con identificar las peticiones que éste necesita para ser invocado. El servicio es representado mediante un Lugar el cual está en estado de espera hasta el momento que recibe una petición. Ésta petición se representa mediante una transición, es aquí donde el servicio procesa la petición y ejecuta la lógica que tiene implementada para enviar una respuesta, la cual también es representada mediante una transición/arista.

Si el servicio que se quiere representar es Web tradicional, como una simple consulta a un diccionario online; o es un servicio Web 2.0, la forma de representarlos es la misma, ya que la diferencia entre estos servicios no está en la tecnología de implementación, ni tampoco en los protocolos usados; la diferencia está en el grado de interacción que tienen con el usuario. Por lo anterior se realizó una representación general para la invocación de servicios Web la cual se ilustra en la Figura 4-7; en la cual se representa la invocación de un servicio Web mediante la petición “HTTP\_Request” y los parámetros necesarios; una vez el servicio realiza la lógica propia con la cual esta implementado, se produce una respuesta, para el presente ejemplo, la cual se representa mediante la respuesta “HTTP\_Response”.

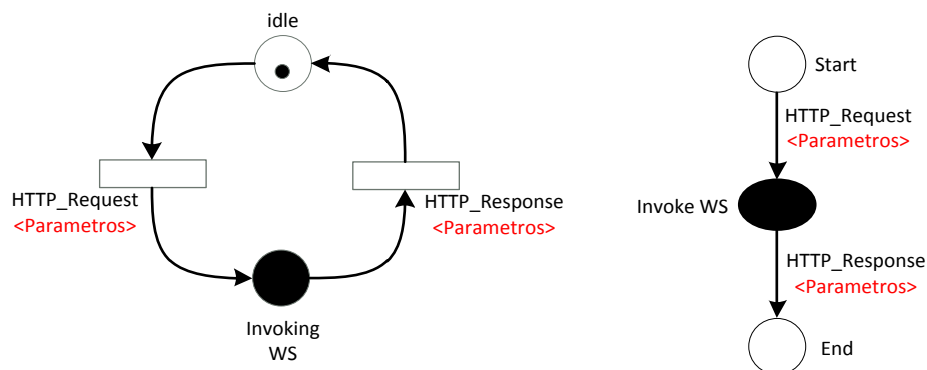


Figura 4-7: Representación genérica de un servicio Web genérico mediante a) redes de Petri b) Grafos

En el caso de que sea necesario tener en cuenta las posibles respuestas del servicios como errores (4XX, 5XX) o respuestas satisfactorias de tipo 200 OK, la representación puede variar un poco como se muestra en la Figura 4-8.



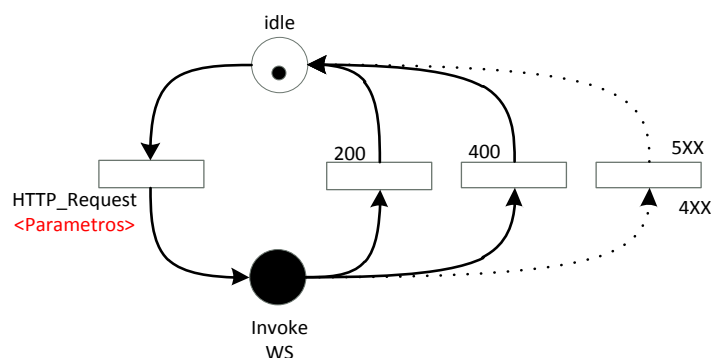


Figura 4-8: Representación de servicio web genérico mediante redes de Petri con posibles respuestas

La descripción, funcionamiento y modelado de diez servicios Web (tradicionales, Web 2.0 y compuestos), se encuentra en el Anexo C.

### 4.2.2.3 Servicios convergentes

Gracias a APIs abiertas para servicios Web como por ejemplo Twitter, Facebook, Ebay, etc., y la fácil invocación de estos mediante peticiones HTTP, los desarrolladores han logrado incorporar las capacidades del mundo Telco, con los nuevos e innovadores servicios Web. Esta integración de mundos ha dado lugar a nuevos servicios convergentes atractivos al usuario final, por esta razón se han modelado un conjunto de servicios, los cuales como se dijo en la Sección 3.4, son aquellos que están compuestos al menos por un servicio Web y un servicio Telco.

El modelado se realizó empleando la representación formal de redes de Petri, como lo muestra el ejemplo de la Figura 4-9, la cual ilustra el servicio Shopping de Mobicents. Este servicio se compone de un servicio Web tradicional de compras y el servicio de llamada con capacidades adicionales de TTS (Text to Speech) y DTMF (Dual Tone Multi Frequency).

El servicio de Shopping implementado por Mobicents en [32], permite al usuario elegir productos en una tienda virtual, agregarlos a un carrito de compras y realizar el pago de los productos; adicionalmente el servicio realiza una llamada de confirmación automática, para corroborar que efectivamente el usuario si ha realizado la compra y su identidad no ha sido suplantada.

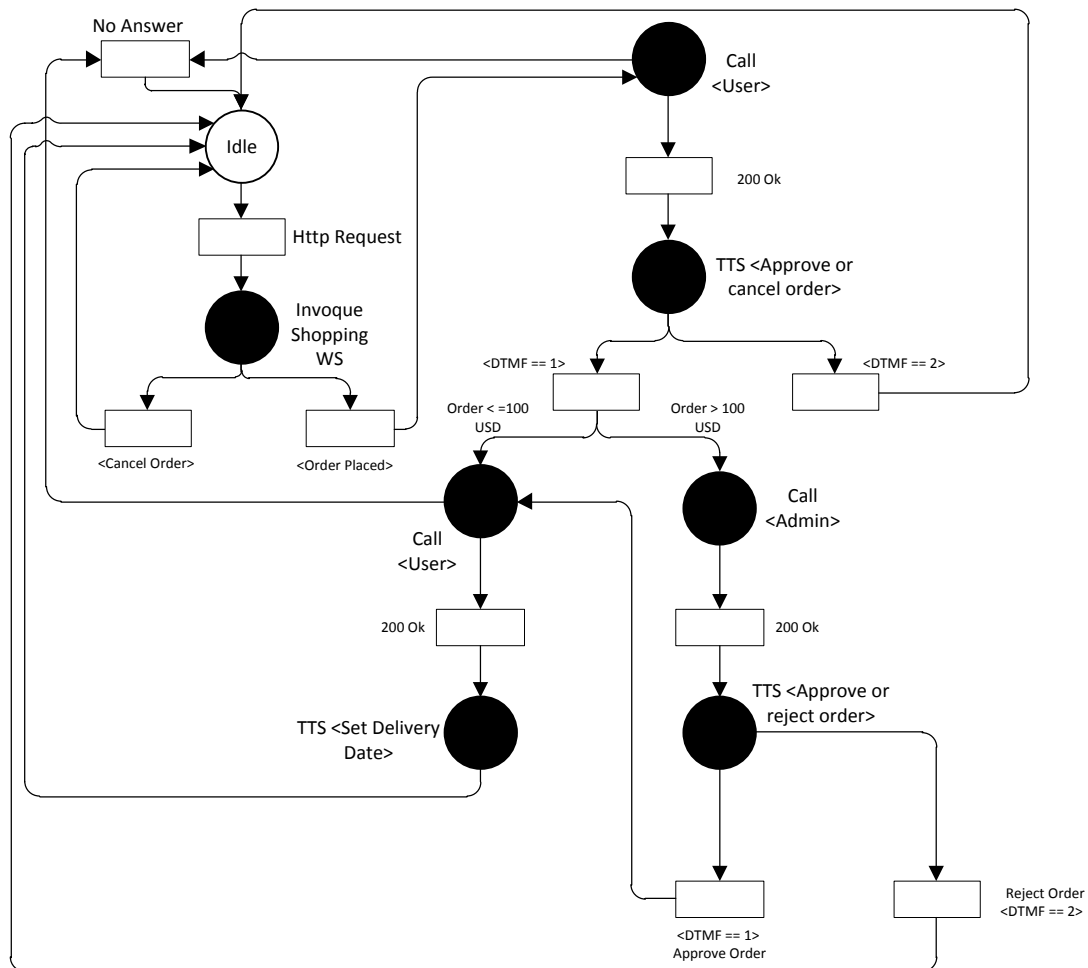


Figura 4-9: Servicio Shopping representado mediante redes de Petri

La descripción detallada del anterior servicio y el modelado de diez servicios Convergentes, se encuentra en el Anexo C.

#### 4.2.2.4 Lista de servicios modelados

La lista final de los servicios modelados (Telco Web y Convergentes) presentes en un entorno Telco 2.0, sobre los cuales se realizará la detección de patrones, se muestra en la Tabla 4-2.

La representación formal en redes de Petri de los treinta servicios se encuentra en el Anexo C.

Tabla 4-2: Servicios modelados en redes de Petri

Servicios Telco	1	Llamada básica
	2	SMS
	3	Llamada + Desvío Incondicional
	4	Llamada + Desvío en Caso de no Respuesta
	5	Llamada en Espera
	6	Llamada en Espera + Transferencia de Llamada
	7	Llamada en Espera + Conferencia
	8	Llamada en Espera + Conferencia + Transferencia de Llamada
	9	Llamada + SMS en caso de ausencia de Respuesta
	10	Llamada + actualización de presencia.
Servicios Web	11	SMS Web
	12	MMS
	13	Ubicación
	14	Capacidades del dispositivo
	15	Pagos
	16	Perfil de conexión de datos
	17	Servicio Skype
	18	Servicio de Twitter
	19	Servicio de Facebook
	20	Amazon – Control de Vendedor
Servicios Convergentes	21	Compras de Mobicents
	22	Mensaje Financiero
	23	Localización + Click2Call
	24	Viajes + Confirmación Sms
	25	Recordatorio de eventos de Facebook con SMS
	26	Recordatorio de eventos de Facebook con SMS y Llamada
	27	EbayProductFinder + Llamada
	28	TwitterSMS (Envío de SMS y Tuits)
	29	CallMessage (Envío de mensajes directos de Twitter + Llamada)
	30	TwitterCalls (Llamada + Envío de SMS + Tuits + Estado de Presencia)

### 4.3 Método para la detección de Patrones de Flujo de ejecución

En ésta sección se describe el proceso realizado para la detección de CFP y DP. Este proceso está basado en el análisis de los servicios modelados en la sección anterior y en un conjunto de posibles patrones presentes en dichos servicios. Al final de esta sección se muestra una lista con los patrones de flujo de ejecución detectados, los cuales servirán como soporte en el proceso de composición de servicios Telco 2.0.

### 4.3.1 Detección de Patrones de Flujo de Control

Para la detección de CFP en servicios Telco 2.0 se propone un método, cuyo principal componente es un algoritmo de detección, el cual permite hacer comparaciones estructurales para determinar similitudes entre grafos. Este módulo se ilustra en la Figura 4-10.

El primer paso del método consiste en transformar los treinta servicios modelados en redes de Petri, en grafos de proceso<sup>14</sup>, los cuales permiten la representación del flujo de control del servicio mediante compuertas lógicas, facilitando así la detección de patrones mediante el algoritmo de detección de subestructuras. El segundo paso consiste en transformar el grafo de proceso en un grafo llamado “Grafo TD” el cual es un archivo de texto con una sintaxis específica que representa el grafo de proceso; y será la base para la detección realizada por el algoritmo. El tercer paso es definir un conjunto de CFP de consulta, los cuales serán detectados en el grafo TD. Finalmente se hace uso de un algoritmo de detección basado en grafos, el cual permite verificar correspondencias entre los CFP de consulta y los CFP presentes en los servicios, dando como resultado una lista de patrones detectados.

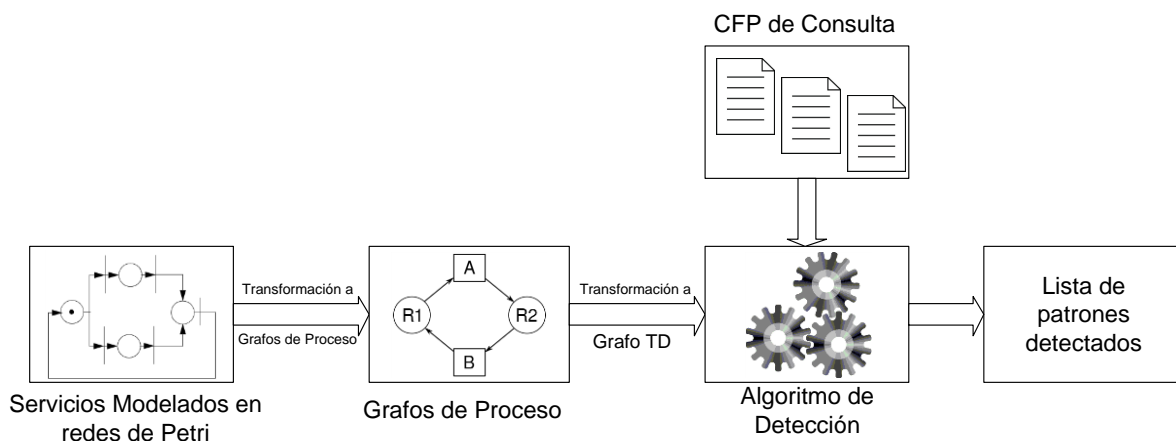


Figura 4-10: Método para la detección de patrones de flujo de control

Cada uno de los componentes de este método se explican en las siguientes secciones.

#### 4.3.1.1 Transformación de redes de Petri a grafos de proceso

Los grafos constituyen una herramienta matemática de modelado formal que utiliza estructuras de datos poderosas y generales para representar objetos y conceptos. Los grafos están constituidos por nodos, que expresan operaciones o cálculos; y aristas o enlaces entre ellos que representan dependencia de datos [64]. De este modo los grafos pueden modelar estructuras con alto nivel de abstracción, y la representación de relaciones de causalidad entre componentes, flujos de datos, flujo de control y concurrencia en la ejecución de procedimientos paralelos.

<sup>14</sup> Grafo de Proceso: Grafo anotado con compuertas lógicas como ANDJ, ORJ XORJ, ANDS, ORJ, XORJ.

Existe un tipo de grafos denominados “grafos de proceso” los cuales utilizan nodos de flujo de control, que pueden tener múltiples entradas y salidas dependiendo de su tipo (XOR (Split-Join), AND (Split-Join) y OR (Split-Join)). De esta manera los flujos de ejecución de un grafo de proceso pueden ser concurrentes y evitar conflictos.

En la Figura 4-11 se puede observar un grafo de proceso en el cual el flujo de datos después del nodo  $E0$  se divide en el nodo de control  $XOR-Split$  y se sincroniza en el nodo de control  $XOR-Join$ . Además entre estos nodos están los nodos básicos  $E1$  y  $E2$  los cuales toman lugar de manera no determinista [65].

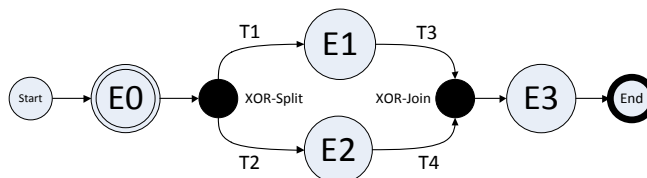
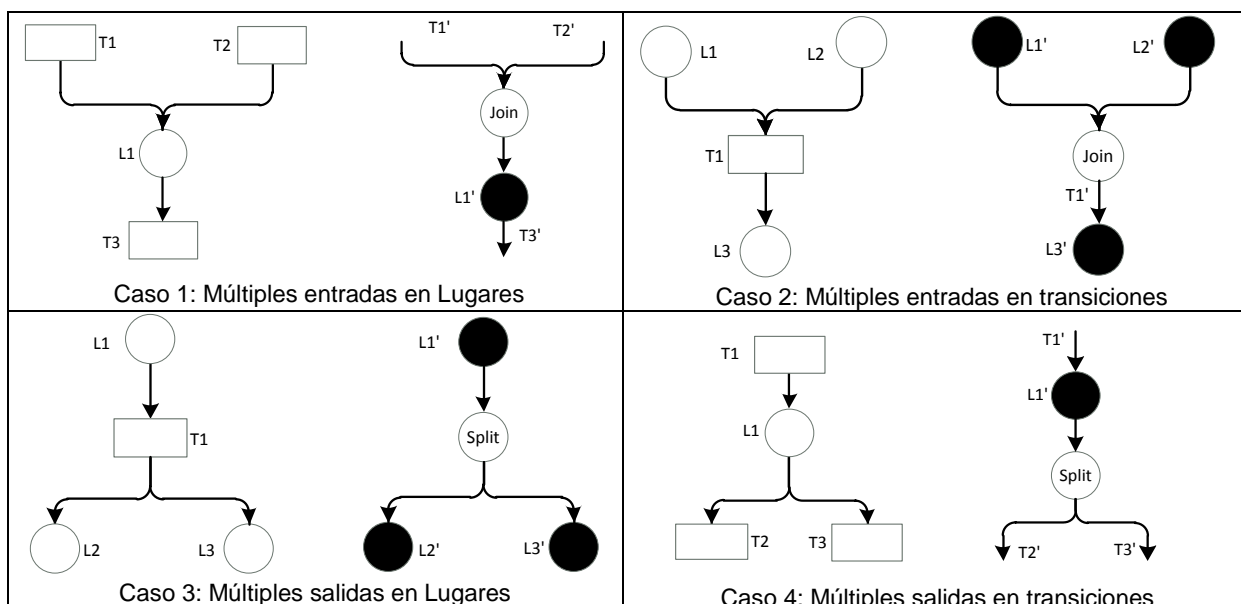


Figura 4-11: Ejemplo de grafo con nodos simples y de flujo de control

En este trabajo se propone una transformación de los servicios modelados, de redes de Petri a grafos de proceso, debido a que estos últimos permiten representar del flujo de control de una forma más específica mediante el uso de compuertas lógicas y es independiente del flujo de datos, adicionalmente permite el uso de un algoritmo de subestructuras el cual facilita el proceso de detección de patrones de flujo de control en los servicios.

Esta transformación se basa en el trabajo de Spiteri [66], donde definen cuatro casos de transformación, los cuales se muestran en la Tabla 4-3.

Tabla 4-3: Transformación de redes de Petri a grafos



En la Figura 4-12 se presenta el servicio básico de *Llamada*, el cual ejemplifica la transformación de redes de Petri a Grafos de proceso, teniendo en cuenta los casos de transformación mostrados anteriormente.

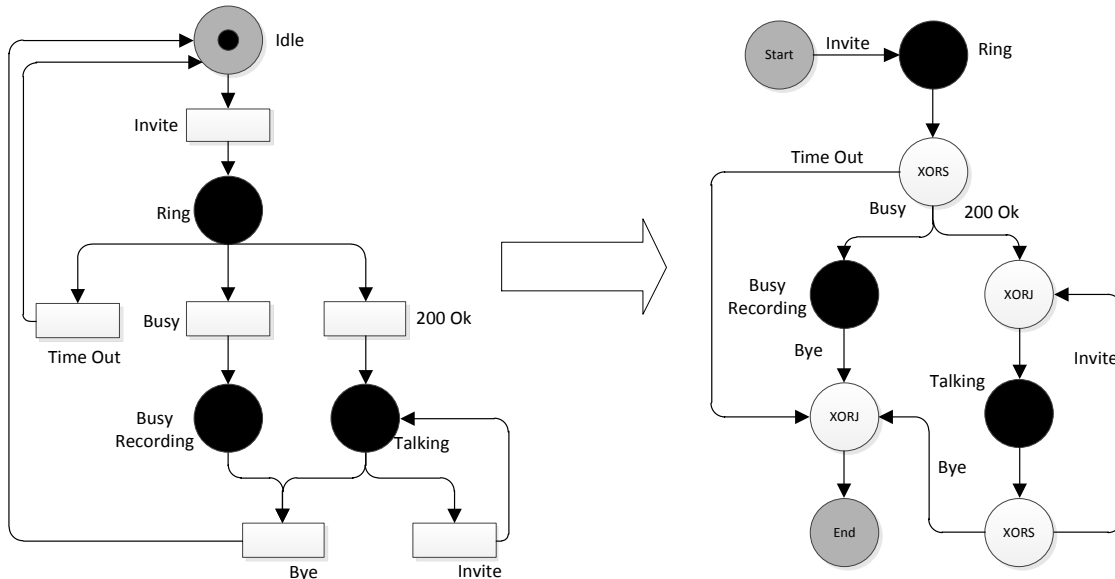


Figura 4-12: Servicio básico de llamada representado mediante redes de Petri (izquierda) y su correspondiente transformación a grafos (derecha)

#### 4.3.1.2 Transformación de Grafos de Proceso a Grafo TD

Este grafo es representado por un archivo con una sintaxis propia, la cual se explica detalladamente en el trabajo de Figueroa [65]. Está compuesto por cinco secciones como se muestra en el grafo de ejemplo de la Figura 4-13 (izquierda). La primera sección es un simple identificador que indica el comienzo de la descripción del archivo (#graph0). La segunda sección, indica el número de nodos que tiene el grafo. La tercera sección, corresponde a las etiquetas del nodo, donde “E”, indica que es un nodo de tipo evento, “T” es un nodo de tipo tarea, y “ORJ”, “ORS”, “XORJ”, “XORS”, “ANDS” Y “ANDJ”, representan los nodos de control. La cuarta sección, corresponde al número de aristas (debe ser al menos 1). Finalmente la quinta sección, corresponde al etiquetado de las aristas, el cual representa el nodo de origen y destino. Ésta transformación es necesaria ya que el algoritmo de detección de CFP (Sección 4.3.1.4) recibe como entrada el archivo descriptor (grafo TD), sobre el cual se realizará la detección de patrones de flujo de control.

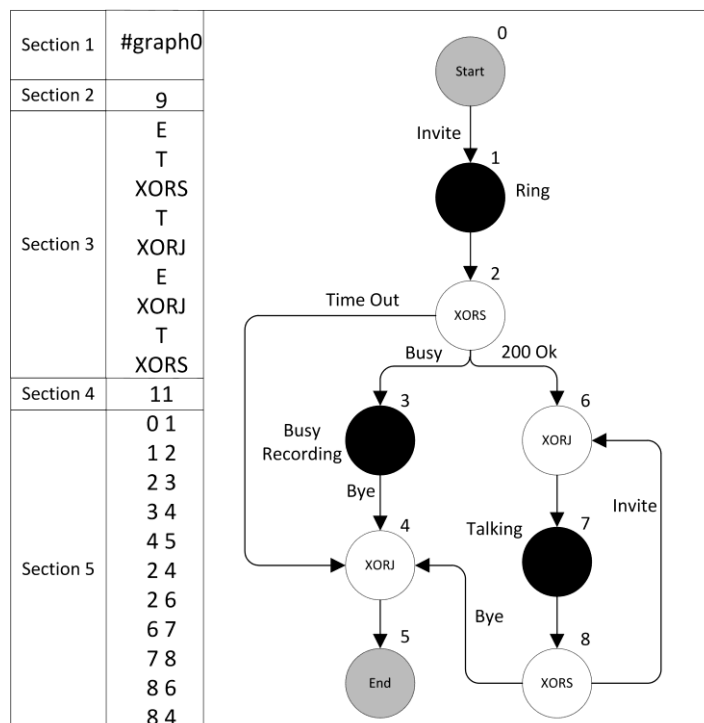


Figura 4-13: Grafo TD del servicio de llamada

### 4.3.1.3 CFP de consulta

El conjunto de patrones de consulta, los cuales son la referencia para el algoritmo de detección, se obtuvo teniendo en cuenta las evaluaciones realizadas por Vander Aalst *et al* [7] [61], a los patrones de flujo de control soportados directamente por los lenguajes JBPM y BPEL ya que estos han sido comúnmente utilizados en procesos de composición de servicios Telco 2.0. Esto se demuestra en algunos trabajos como [4] [5] [6]. La evaluación de los anteriores lenguajes se presenta en el Anexo C,

Adicionalmente se determinó un criterio de selección de patrones el cual fue: “*El patrón de objeto de estudio debe ser soportado directamente por el lenguaje BPEL o JBPM*”: esto significa que el patrón debe ser soportado al menos por uno de estos lenguajes, por ejemplo: el patrón *Simple Merge* es soportado por ambos lenguajes, por lo tanto, se define como un patrón de consulta; el patrón *Deferred Choice* es únicamente soportado por JBPM, por lo cual, también es considerado como patrón de consulta; finalmente los patrones que no sean soportados por ninguno de los dos lenguajes, no se tienen en cuenta para este grupo.

El resultado de esta selección fue un conjunto de 21 CFP los cuales se muestran en la Tabla 4-4.

Una vez escogido un grupo de patrones de consulta, estos pueden ser clasificados según su funcionalidad y según su comportamiento; ambas clasificaciones se explican a continuación.

- **Clasificación de patrones según su funcionalidad:** Las categorías definidas para la clasificación de patrones según su funcionalidad son: AND, XOR, OR y SEQ. La primera de ellas denominada AND, hace referencia a un funcionamiento síncrono del

flujo de control. La segunda categoría es denominada XOR y hace referencia a un funcionamiento asíncrono del flujo de control, es decir que un patrón que permita generar asincronía en el flujo de control. La tercera categoría es denominada OR y su funcionamiento consiste en que de acuerdo a una serie de condiciones establecidas, puede permitir la sincronía o asincronía del flujo de control. Finalmente, La cuarta categoría es denominada SEQ, y hace referencia a un funcionamiento secuencial del flujo de control en el cual no se presenta sincronía ni asincronía.

- **Clasificación de patrones según su comportamiento:** las categorías definidas para la clasificación de patrones según su comportamiento son SPLIT, JOIN y KEEP. La primera de estas tres categorías denominada SPLIT hace referencia a los patrones que presentan un comportamiento de divergencia en el flujo de control en 2 o más ramas, en contraste con la categoría ya mencionada, la segunda categoría denominada JOIN, hace referencia a los patrones que presentan un comportamiento de convergencia en el flujo de control de 2 o más ramas; finalmente, la tercera categoría denominada KEEP, hace referencia a los patrones que no realizan divergencia ni convergencia del flujo de control, es decir, aquellos que mantienen la coherencia del flujo de control.

Los CFP y su clasificación de acuerdo al comportamiento y funcionalidad, se resumen en la Tabla 4-4; es importante aclarar que ambas clasificaciones se denotan con una sola palabra, por ejemplo: la funcionalidad AND y comportamiento SPLIT corresponde con ANDS. Cada una de estas palabras corresponde a su vez a una compuerta lógica, lo cual facilita el entendimiento y representación formal de cada patrón, sin embargo, existen patrones que necesitan más de una compuerta lógica para su representación.

Tabla 4-4: Conjunto de patrones de objeto de estudio clasificados según su funcionalidad y comportamiento

	<b>Patrón</b>	<b>Compuerta</b>	<b>Comentarios</b>
<b>1</b>	Sequence	-	
<b>2</b>	Parallel Split	ANDS	
<b>3</b>	Synchronization	ANDJ	
<b>4</b>	Exclusive Choice	XORS	
<b>5</b>	Simple Merge	XORJ	
<b>6</b>	Multi-choice	ORS	
<b>7</b>	Structured synchronizing merge	ORS + ANDJ	
<b>8</b>	Multi-merge	ORJ	
<b>9</b>	Generalized AND-join	ANDS + ANDJ	
<b>10</b>	Local synchronizing merge	ORS + ANDJ	Se necesita sincronizaciones parciales o locales del flujo mediante compuertas XORJ y XORS. Este patrón es una variación de representación del <i>Structured synchronizing merge</i>
<b>11</b>	Deferred Choice	XORS	
<b>12</b>	Critical Section	ANDS + ANDJ	
<b>13</b>	Interleaved Routing	ANDS + ANDJ	



14	Multiple instances without synchronization	XORS	
15	Cancel Task	XORS ó XORJ	Una rama, debe ir a un nodo de cancelación. La rama de destino de XORJ debe ir al nodo de cancelación
16	Cancel Case	XORS ó XORJ	Una rama debe ir a un nodo de fin La rama de destino de XORJ debe ir al fin
17	Arbitrary Cycles	XORS + XORJ	Una rama de la compuerta XORS es dirigida a la compuerta XORJ.
18	Structured Loop	XORS +XORJ	Una rama de la compuerta XORS es dirigida a la compuerta XORJ.
19	Implicit Termination	ANDS	
20	Transient Trigger	ANDJ	
21	Persistent Trigger	ANDJ	

La explicación detallada del funcionamiento y comportamiento de cada uno de los 21 CFP además de su representación mediante redes de Petri y Grafos se encuentra en el Anexo A.

#### 4.3.1.4 Algoritmo para la detección de CFP

La detección de CFP en servicios Telco 2.0, se realiza mediante un proceso de detección de subestructuras basado en isomorfismo de grafos<sup>15</sup> [67]. Existen algunos algoritmos que abordan la detección de grafos isomórficos, algunos de los más sobresaliente según [68] son: Ullmann, Nauty, Schmidt & Druffel (SD), VF, VF2. Los cuales son comparados utilizando un banco de pruebas de 10.000 parejas de grafos isomórficos y se concluye que el algoritmo VF2 marca una diferencia notoria sobre los otros siendo este más eficiente en cuanto a tiempo de respuesta, fiabilidad y requerimientos de memoria. A ésta misma conclusión llegan los trabajos [67] y [69].

Teniendo en cuenta las ventajas del algoritmo VF2, existen herramientas software que utilizan este algoritmo para realizar comparaciones entre grafos como GraphBlast [70] y NetMatch [71]; el primero de éstos realiza comparaciones con nodos no dirigidos y el segundo es utilizado para la búsqueda de sistemas biológicos como personas, cadenas de ADN y ARN, con grafos dirigidos.

Para el presente trabajo de grado se tuvo en cuenta el proyecto de investigación de los Ingenieros Rivas y Corchuelo [72], los cuales utilizaron la herramienta GraphBlast y el algoritmo VF2 para la detección de patrones de flujo de control en procesos de negocio semánticamente enriquecidos. El prototipo funcional de dicho trabajo fue adaptado con el fin de incluir los 21 CFP de consulta descritos en la Sección 4.3.1.3, que son la referencia

<sup>15</sup> Grafos Isomórficos: dos grafos son isomórficos si tienen el mismo número de nodos y la misma estructura.

utilizada por el algoritmo de detección para encontrar patrones de flujo de control en los servicios modelados.

La Figura 4-14 muestra un esquema del proceso de detección. Como se observa, el módulo de detección utiliza el grafo TD, para buscar correspondencias con los 21 patrones de consulta, en caso de encontrarlas, el algoritmo arroja como resultado el patrón detectado y los nodos donde se encontró dicho patrón.

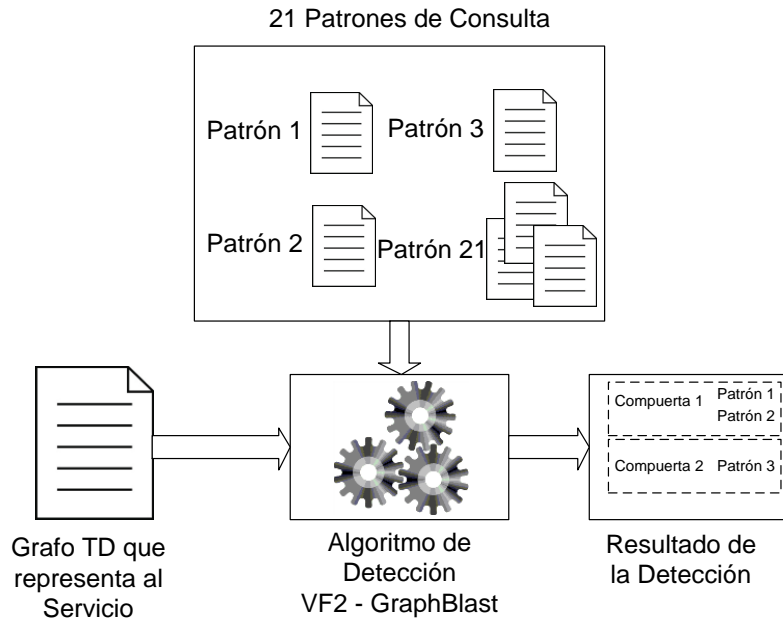


Figura 4-14: Proceso de detección de patrones de flujo de ejecución

De esta forma el algoritmo de detección de subestructura utilizado en este trabajo, indica los puntos críticos del servicio los cuales representan la existencia de un patrón de flujo de control. Dichos puntos son representados mediante compuertas lógicas y asociados a un CFP de acuerdo a la Tabla 4-4.

Debido a que algunos CFP son representados por las mismas compuertas lógicas, por ejemplo: *Generalized AND-join*, *Critical Section* y *Interleaved Routing*, se utiliza el modelo en redes de Petri para determinar el CFP resultante. Para los CFP representados con una sola compuerta como *Multi Merge (ORJ)* o *Multi-Choice (ORS)* no es necesario realizar ésta validación.

El proceso de detección CFP en los treinta servicios presentados en la sección 4.2.2, dio como resultado un conjunto de 13 patrones detectados. Esto se explica en detalle en la Sección 4.4.1.

La Figura 4-15 muestra un ejemplo de detección, en la cual se define un grafo con 6 nodos, los cuales representan el flujo de control del servicio; y tres de los 21 patrones de consulta: *Multi Merge*, *Multi-Choice* y *Simple Merge*. El algoritmo hace un recorrido del Grafo que representa al servicio, para buscar correspondencias con los patrones de consulta. Para el

ejemplo, el algoritmo detecta el patrón *Multi-Choice* en los nodos 2 -3 y 4, y el patrón *Multi Merge* en los nodos 3- 4 y 5. El patrón de consulta *Simple Merge*, no es detectado.

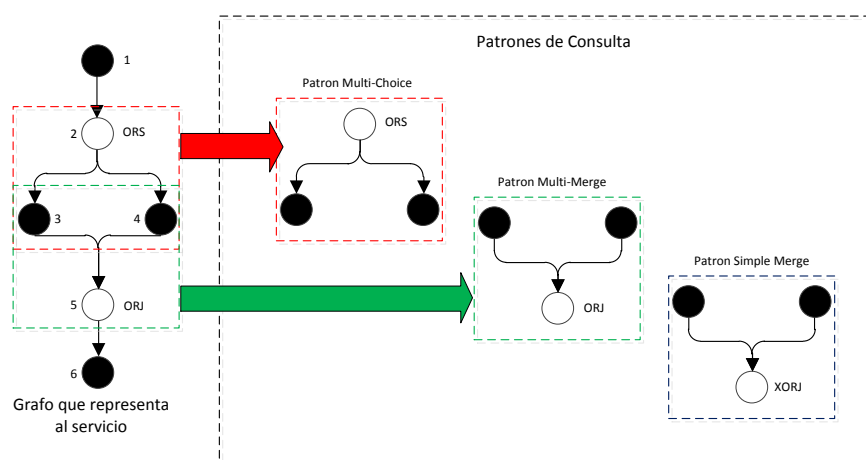


Figura 4-15: Ejemplo de detección de patrones de flujo de control

### 4.3.2 Detección de Patrones de Datos

Como se dijo al inicio de este Capítulo los DP tienen como objetivo común, capturar las diversas formas en las cuales se representan los datos. Estos patrones son independientes del lenguaje de modelado y de la tecnología utilizada.

Del conjunto de DP definido por Vander Aalst *et al* [8] se escogió un subconjunto de estudio, teniendo en cuenta la evaluación realizada en [8] y [73], de los lenguajes JBPM y BPEL, los cuales son utilizados para la composición de servicios en los trabajos [4] [5] [6]. La condición para que el patrón entre a este grupo es la siguiente: “El patrón de objeto de estudio debe ser soportado por el lenguaje BPEL o JBPM”. Por ejemplo el patrón *Data Interaction between Tasks* es soportado por BPEL y JBPM simultáneamente, por lo cual, ingresa al grupo de estudio; el patrón *Scope Data*, solo es soportado por BPEL, sin embargo, también ingresa al grupo de estudio; y finalmente el resto de DP que no son soportados por ningún lenguaje, no se tienen en cuenta para su estudio. Los resultados de ésta evaluación son presentados en el Anexo B.

Gracias al proceso explicado anteriormente, el conjunto de patrones de estudio se reduce de 40 a 16. Una explicación detallada del funcionamiento y comportamiento de este conjunto de patrones, además de su representación mediante redes de Petri se encuentra en el Anexo B.

Por otra parte, para el conjunto de patrones de objeto de estudio, se hizo una revisión de la especificación JAIN SLEE [17] con el fin de evaluar la viabilidad de implementación de los DP. La Tabla 4-5 muestra los patrones y la forma como JAIN SLEE los soporta.

Tabla 4-5: Patrones de datos y su correspondiente implementación con JAIN SLEE

		<b>Patrón [8]</b>	<b>Soporte JAIN SLEE</b>
Visibilidad de Datos	1	Scope Data	<ul style="list-style-type: none"> <li>• <b>Slee Profiles:</b> Define los elementos de un servicio en relación al perfil de cada usuario [74].</li> </ul>
	2	Case Data	
	3	Environment Data	
Interacción Interna de Datos	4	Data Interaction between Tasks	<ul style="list-style-type: none"> <li>• <b>Firing Events:</b> Disparo de eventos dentro del Slee [17].</li> <li>• <b>Activity Context y Activity objects:</b> objetos java dentro del Slee, y su definición.</li> </ul>
Interacción Externa de Datos	5	Data Interaction - Task to Environment - Push-Oriented	<ul style="list-style-type: none"> <li>• <b>Firing Events:</b> Disparo de eventos hacia y desde procesos y aplicaciones externas.</li> <li>• <b>Adaptadores de Recurso</b></li> </ul>
	6	Data Interaction - Environment to Task - Pull-Oriented	
	7	Data Interaction - Task to Environment - Pull-Oriented	
Transferencia de Datos	8	Data Passing by Value - Incoming	<ul style="list-style-type: none"> <li>• <b>Adaptadores de Recurso:</b> Permiten la interacción del SLEE con aplicaciones, servicios y entornos externos; algunos de estos son Http-Servlet RA, Slp RA , etc.</li> <li>• <b>Integración con bases de datos (JPA, DAO).</b></li> <li>• <b>Activity Context y Activity objects.</b></li> </ul>
	9	Data Passing by Value - Outgoing	
	10	Data Passing - Copy In/Copy Out	
	11	Data Passing by Reference - Unlocked	
	12	Data Transformation - Input	
	13	Data Transformation - Output	
Enrutamiento de Datos	14	Task Precondition - Data Value	<ul style="list-style-type: none"> <li>• <b>Events Routing:</b> Eventos disparados desde aplicaciones externas.</li> <li>• <b>Firing Events</b></li> </ul>
	15	Event-based Task Trigger	
	16	Data-based Routing	

Finalmente, se puede concluir que el conjunto de DP mostrados en anterior estudio, permiten capturar el paso de datos provenientes desde: el interior del SLEE, aplicaciones externas, bases de datos, etc., dando soporte al proceso de composición de servicios Telco 2.0.

## 4.4 Patrones de Flujo de Ejecución Presentes en servicios Telco 2.0

### 4.4.1 Lista de Patrones de Flujo de Control presentes en un entorno Telco 2.0

El método para la detección de CFP el cual se explicó en las secciones previas, fue aplicado para cada uno de los treinta servicios que se encuentran descritos en el Anexo C.

Una vez obtenido el número de CFP detectados de cada servicio, se procedió a obtener la recurrencia de cada patrón teniendo en cuenta el total detectado, utilizando la ecuación (1):

$$\text{Recurrencia patrón } n = \frac{\text{Total } p_n}{P} \times 100 \quad (1)$$

$$\therefore n = \text{patrón } \{1,2,..21\}$$

$$p_n = \text{cantidad de patrones identificados de } n$$

$$P = \text{Total de Patrones identificados}$$

Con la ecuación (1) se obtuvieron los resultados que se muestran en la Figura 4-16. Como se puede observar los patrones más recurrentes son: *Simple Merge* (P5) el cual converge de manera asíncrona el flujo de control, *Deferred Choice* (P11) el cual diverge el flujo dependiendo de un estado previo, y *Cancel Case* (P16) el cual representa la terminación de actividades dentro del flujo; y los que obtuvieron menos recurrencia en el conjunto de servicios modelados fueron: *Multi Choice* debido a que este patrón puede ser compuesto por la conjunción de los patrones *Parallel Split* y *Exclusive Choice*; *Generalized And-Join*, entre otros. Por otra parte, no se detectaron los patrones P7, P10, P12, P13, P14, P19, P20, y P21.

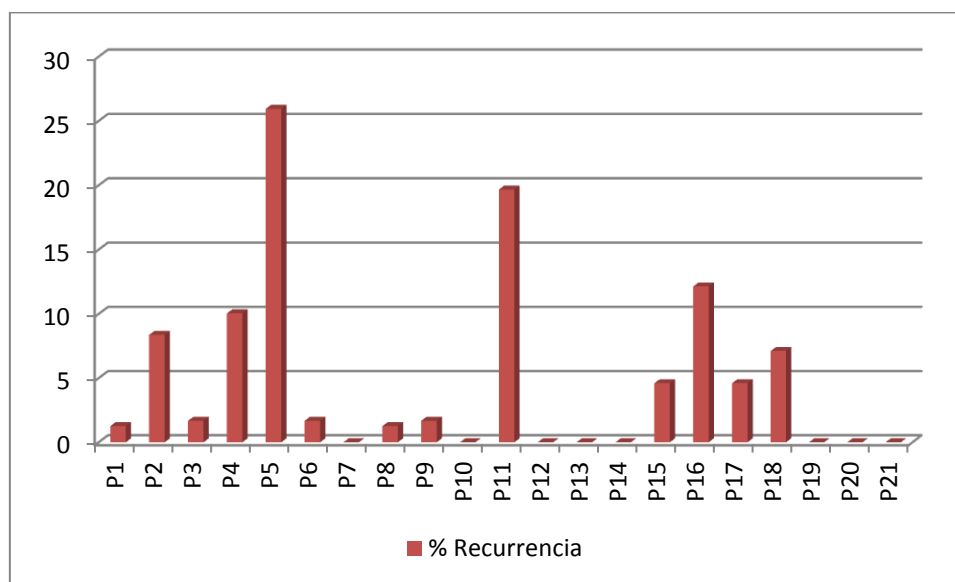


Figura 4-16: Resultados de la detección de patrones

Gracias a los resultados obtenidos, el número de patrones de estudio se reduce de 21 a 13, debido a que 8 de ellos no fueron detectados en el conjunto de servicios modelados.

Es importante resaltar que los patrones P1 a P5 son patrones básicos los cuales son la base para la formación del conjunto de 43 patrones definidos en [7]. Finalmente los 13 patrones identificados se resumen en la Tabla 4-6, los cuales son implementados en la herramienta **GT-4SC**, la cual permite componer servicios convergentes de manera gráfica, mediante dichos patrones.

Tabla 4-6: Patrones identificados después de la detección de patrones

ID	Patrón
P1	Secuence
P2	Parallel Split
P3	Synchronization
P4	Exclusive Choice
P5	Simple Merge
P6	Multi-Choice
P8	Multi-Merge
P9	Generalized And-Join
P11	Deferred Choice
P15	Cancel Task
P16	Cancel Case
P17	Arbitrary Cycles
P18	Structured Loop

#### 4.4.2 Lista de patrones de datos presentes en un entorno Telco 2.0

Teniendo en cuenta el estudio realizado en la Sección 4.3.2, en la cual se identificaron los patrones de datos soportados por JAIN SLEE, la Tabla 4-7 presenta la lista final de DP aplicables para la implementación de servicios Telco 2.0.

Tabla 4-7: Lista de patrones de datos presentes en un entorno Telco 2.0

ID	Patrón
1	Scope Data
2	Case Data
3	Environment Data
4	Data Interaction between Tasks
5	Data Interaction - Task to Environment - Push-Oriented
6	Data Interaction - Environment to Task - Pull-Oriented
7	Data Interaction - Task to Environment - Pull-Oriented
8	Data Passing by Value - Incoming
9	Data Passing by Value - Outgoing
10	Data Passing - Copy In/Copy Out

11	Data Passing by Reference - Unlocked
12	Data Transformation - Input
13	Data Transformation - Output
14	Task Precondition - Data Value
15	Event-based Task Trigger
16	Data-based Routing

#### 4.4.3 Ejemplo – composición de un servicio Telco 2.0 mediante patrones de flujo de ejecución

Para demostrar los resultados de la detección de patrones de flujo control y la viabilidad de implementación de los patrones de datos en JAIN SLEE, se utiliza como ejemplo demostrativo el servicio “CallMessage” (Llamada + Mensaje directo de twitter), el cual permite a un usuario A, hacer una llamada desde una aplicación Web a un usuario B; en caso de que este último no conteste, se envía un mensaje directo de Twitter a B, con la información de que A se intentó comunicar con él.

La representación de este servicio en grafos de proceso, y los CFP detectados se muestran en la Figura 4-17.

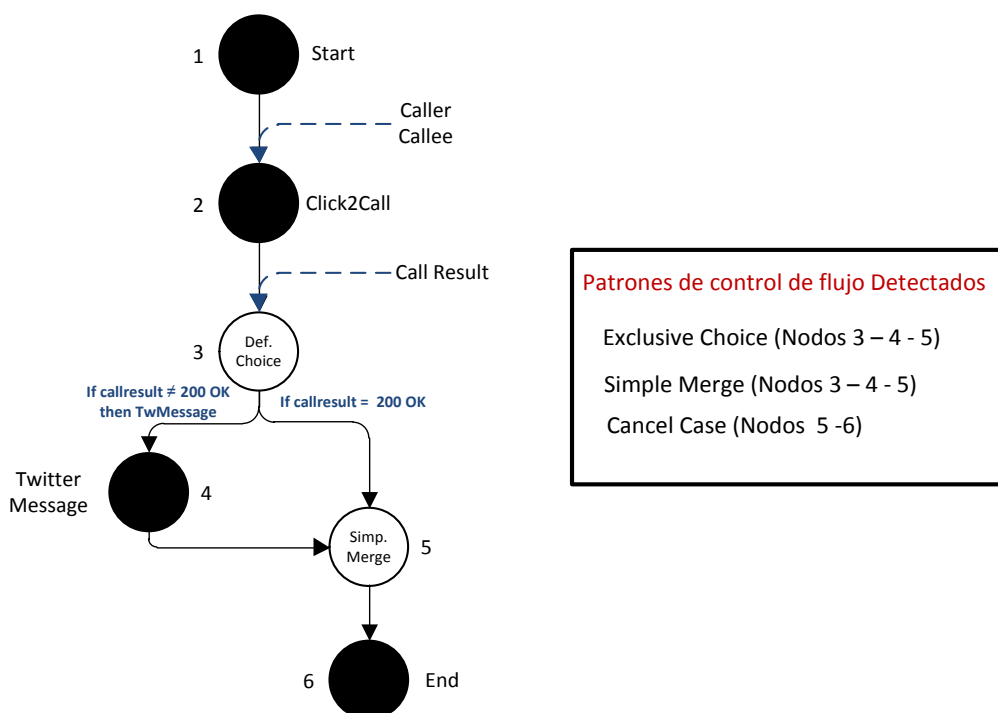


Figura 4-17: Ejemplo de detección de patrones de flujo de ejecución en el servicio “CallMessage”

Como se observa en la Figura 4-17 la composición del servicio CallMessage es soportada por los patrones *Exclusive Choice*, *Simple Merge* y *Cancel Case*.

Por otra parte, para la implementación de este servicio en JAIN SLEE, se presentan los siguientes requerimientos.

- La aplicación Web debe enviar un evento al SLEE para que éste inicie la llamada, con la información del usuario emisor y el destinatario.
- SLEE debe enviar un evento a los usuarios con la invitación a hablar.
- En caso de que el usuario B no conteste, el SLEE debe enviar un evento de cancelación, el cual dispare el servicio Web que envía el mensaje directo de Twitter
- Se debe buscar en la base de datos de usuario el screenID de Twitter <sup>16</sup> para enviar el mensaje.
- En caso de que el mensaje sea enviado satisfactoriamente, el servicio Web debe enviar un evento al SLEE de confirmación.

Los anteriores requerimientos, se representan como patrones de datos los cuales soporta JAIN SLEE de la siguiente forma:

- La transferencia de datos desde una aplicación externa, la cual es representada por los patrones de *Transferencia de datos*, JAIN SLEE los soporta con el uso de adaptadores de recurso, los cuales reciben información de la red IP; y además con los *disparos de eventos (Firing events)*, definidos en la especificación [17].
- La transferencia de datos desde bases de datos, la cual es representada por el patrón *copia de datos*, es soportada por JAIN SLEE, con el uso de conectores como JDBC.
- La transferencia de datos al interior del SLEE, la cual es representada por el patrón *Interacción entre datos*, JAIN SLEE los soporta con la definición de *objetos de actividad y contextos de actividad* [17].

## 4.5 Resumen

Este Capítulo presentó un método para la detección de patrones de flujo de ejecución, el cual se divide en la detección de CFP y DP.

Para la detección de CFP, el método consiste en modelar un conjunto de treinta servicios presentes en el catalogo propuesto en la Sección 3.4, mediante la representación formal de redes de Petri. A continuación se transforma la red de Petri en Grafos de proceso, y estos a su vez a la sintaxis propia del “Grafo TD”. Por otra parte, para la definición de los patrones de consulta utilizados por el algoritmo de detección, se realizó un estudio de los CFP soportados por JBPM y BPEL, lo cual redujo el número de 43 patrones a 21. Finalizados los anteriores pasos, se utiliza el algoritmo de detección de subestructuras el cual busca similitud de nodos entre el grafo TD y los CFP para determinar los puntos críticos donde hay existencia de patrones. Después, se compara la red de Petri con el servicio para establecer los nodos que corresponden a cada patrón. Finalmente este proceso da como resultado un conjunto de 13 CFP los cuales serán implementados en la herramienta GT-4SCE (Capítulo 5).

Por otra parte, para la detección de DP, se hizo un estudio basado en las evaluación de Vander Aalst *et al.* de los DP soportados por JBPM y BPEL, lo cual redujo el número de

---

<sup>16</sup> ScreenID de Twitter: Es el valor numérico que representa a un usuario de Twitter



patrones de 40 a 16. Para cada uno de estos 16 DP se estudió la viabilidad de implementación según la especificación JAIN SLEE [17]. Finalmente, los 16 patrones soportados por JAIN SLEE serán utilizados en la herramienta GT-4SCE (Capítulo 5).



## Capítulo 5

### 5 Prototipo, Experimentación y Resultados

El presente Capítulo busca evaluar experimentalmente a través de un prototipo, la composición de servicios convergentes basada en los patrones del flujo de ejecución detectados en servicios Telco 2.0 en el Capítulo 4. Para esto se exponen dos secciones principales: la primera hace referencia al desarrollo de una herramienta que permite componer servicios Telco 2.0 de forma gráfica; y la segunda corresponde una fase de evaluación, pruebas y experimentación del prototipo propuesto, con el fin de validar si la composición de servicios Telco 2.0 basada en patrones de flujo de ejecución, reduce el tiempo del proceso de composición, en comparación a realizar el mismo proceso sin el soporte de dichos patrones.

#### 5.1 Prototipo Funcional

A continuación se describe el prototipo realizado en este trabajo de investigación, el cual busca ofrecerle al usuario una herramienta gráfica para componer servicios Telco 2.0 basada en patrones de flujo de ejecución. Esta herramienta permite crear nuevos servicios mediante el uso de servicios básicos previamente implementados, los cuales serán desplegados y ejecutados en un servidor JAIN SLEE con el fin de probar su funcionamiento

Adicionalmente, se muestra la arquitectura de la herramienta y los diagramas de despliegue, orquestación y paquetes correspondientes al entorno de desarrollo y ejecución; finalmente se describen tres servicios de prueba creados para evaluar el funcionamiento del prototipo.

##### 5.1.1 Descripción de la herramienta GT-4SC (Graphical Tool-patterns-based for Service Creation)

GT-4SC es una extensión (Plug-in) del entorno de desarrollo integrado Eclipse [75], desarrollado e implementado en el marco del presente trabajo de investigación, el cual permite componer servicios Telco 2.0 a través de una interfaz gráfica compuesta por módulos reutilizables. Cada módulo tiene un código fuente asociado, que se genera automáticamente al agregar uno de estos al área de trabajo. Esta capacidad le facilita al desarrollador, la composición de nuevos servicios evitando el contacto con el código fuente y la configuración de archivos complementarios de funcionamiento.

GT-4SC es una herramienta para la composición de servicios Telco 2.0 basada en patrones de flujo de ejecución, lo cual permite al usuario implementar nuevos servicios teniendo en cuenta las perspectivas de flujo de control y datos.

La interfaz gráfica de GT-4SC se ilustra en la Figura 5-1 y está compuesta por los elementos que se describen a continuación.

- *Elementos básicos*: son elementos complementarios a la herramienta, que permiten el acceso a funcionalidades como selección, y movimiento de componentes dentro del área de trabajo.
- *Eventos de inicio y fin*: especifican el inicio y fin de un servicio. Cuando estos componentes son agregados al área de trabajo, se genera automáticamente el código básico de configuración necesario para trabajar en el entorno de desarrollo.
- *Servicios básicos disponibles*: estos componentes están formados por servicios básicos previamente implementados, los cuales pueden ser reutilizados dependiendo de la funcionalidad requerida por un servicio. Una vez agregados al área de trabajo, el código necesario para cada servicio es generado automáticamente, dicho código incluye las referencias necesarias, definición de eventos de entrada y salida, generación de métodos para recepción y emisión de eventos, etc.
- *Patrones de flujo de control*: estos módulos permiten definir el flujo de control del servicio, lo cual se hace a través de CFP previamente definidos.
- *Conector*: permite conectar módulos dentro del área de trabajo.

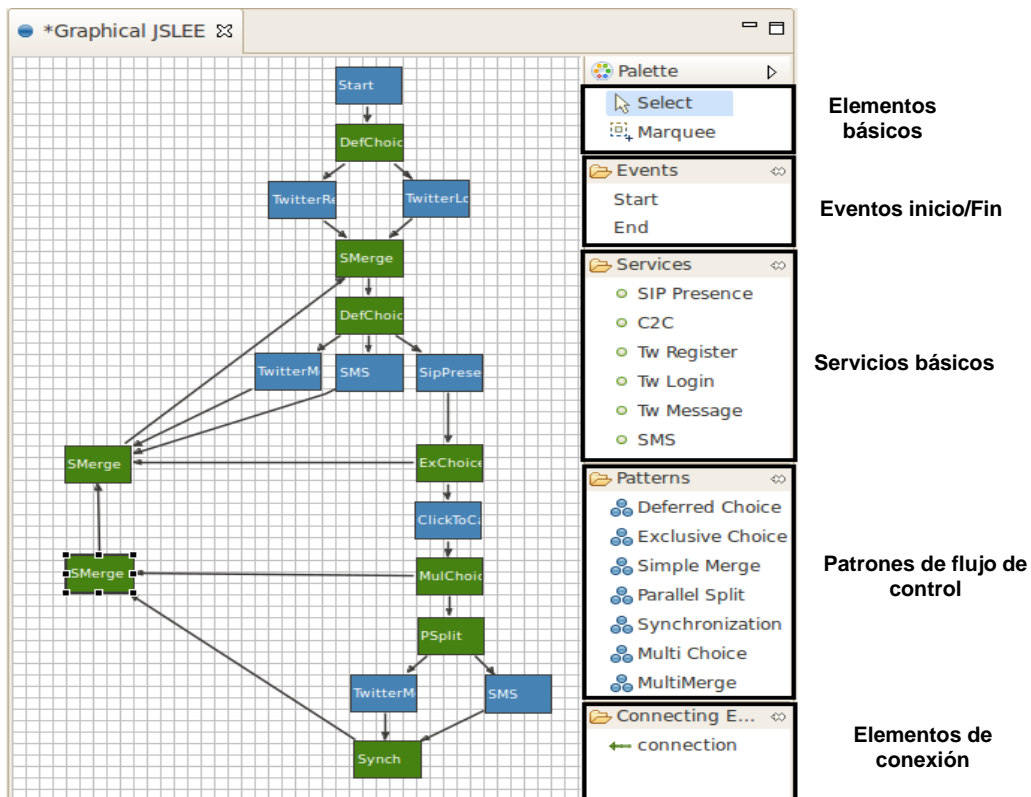


Figura 5-1: Módulos de la herramienta GT-4SC

## 5.1.2 Diagramas y Arquitectura

En esta sección se muestra la arquitectura correspondiente a la herramienta GT-4SC, así como los diagramas que describen el funcionamiento del prototipo.

### 5.1.2.1 Arquitectura de la Herramienta

La arquitectura de la herramienta GT-4SC está enmarcada dentro del desarrollo de una extensión de Eclipse y la componen un conjunto de herramientas software como: GEF, JET y algunos módulos adicionales, los cuales se ilustran en la Figura 5-2 y se describen a continuación:

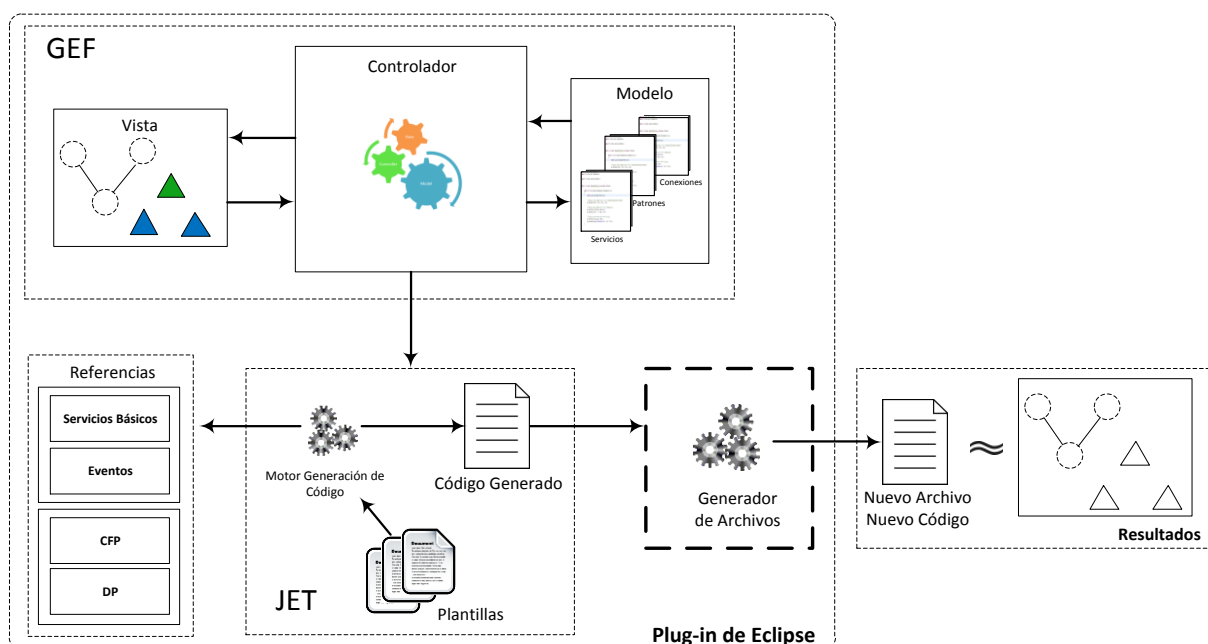


Figura 5-2: Arquitectura de la herramienta

**Plug-in de Eclipse:** módulo software o bloque constructor dentro de la plataforma Eclipse. Utiliza las librerías y complementos necesarios para ser la base de implementación de la herramienta GT-4SC. Este Plug-in está compuesto por elementos como un editor gráfico, paleta de elementos, menús emergentes, entre otros; además utiliza los frameworks GEF y JET con el fin de implementar las funcionalidades gráficas y de generación de código requeridas.

**GEF (Graphical Editing Framework):** este framework provee la tecnología necesaria para crear vistas y editores gráficos enriquecidos dentro del área de trabajo de Eclipse IDE. GEF sigue el patrón de programación MVC (modelo, vista y controlador), lo cual se aplica a esta arquitectura como se explica a continuación.

- *Vista:* bloques constructores que componen la interfaz gráfica, en la cual se muestra la información de la aplicación. En este caso estos bloques representan cada servicio, los CFP y las conexiones.

- *Modelo*: corresponde a los datos de la aplicación, en este caso el modelo representa las características y atributos de los patrones, servicios y aristas.
- *Controlador*: Implementa la lógica y es el enlace entre el modelo y la vista.

**JET (Java Emitter Template)**: JET es un motor “Modelo a Texto” que permite la generación de código a través del uso de una tecnología basada en plantillas, las cuales manejan una sintaxis similar a la utilizada por Java Server Pages (JSPs). Este módulo está compuesto por los siguientes elementos:

- *Plantillas*: ficheros con extensión .jet que contienen contenido predefinido, los cuales serán utilizados por el motor de generación de código (Code Generator Engine) para la generación de porciones de código o snippets.
- *Motor de Generación de Código*: está basado en un proceso llamado “generación”, el cual consiste en generar clases java de implementación a partir de las plantillas (templates) definidas.
- *Código Generado*: es el código resultante generado por el motor de generación de código (Code Generator Engine) a partir de las plantillas.

### Módulos Adicionales

- **Referencias**: este módulo corresponde a las referencias de ubicación de las unidades desplegadas de los servicios básicos y sus respectivos eventos de inicio y respuesta. Estos servicios y eventos están almacenados dentro de la herramienta GT-4SC y son referenciados en la creación de un nuevo servicio a través del resultado emitido por el generador de código. Los componentes principales de este módulo están constituidos por los patrones de flujo de control y datos; estos se explican a continuación.
- **Componente CFP**: comprende la referencia a los patrones de flujo de control dentro de la herramienta; esta referencia es utilizada por el generador de código para implementar cada patrón de acuerdo a los requerimientos de un servicio. La lista de CFP soportados por GT-4SC y una comparación con los lenguajes JBPM y BPEL, se encuentra en el Anexo E.
- **Componente DP**: comprende la referencia a los patrones de datos dentro de la herramienta; esta referencia es utilizada por el generador de código para implementar cada patrón de acuerdo a los requerimientos de un servicio. La lista de DP soportados por GT-4SC y una comparación con los lenguajes JBPM y BPEL, se encuentra en el Anexo E.

El código fuente asociado a los módulos CFP y DP se muestra en la Figura 5-3, dicho código comprende una función asociada a cada CFP el cual es implementado dependiendo de la llegada de datos específicos. Cada dato de llegada representa el uso de un DP y define la implementación de la función de control (ej: llegada de datos externos o internos, disparo de un evento dependiendo del valor de una variable interna, etc.).

```

/** Función Flujo de Control y Flujo de Datos */
Funcion FlujoControl (Parametros FlujoDatos)
{
  /** Implementación Función */
}

/** llamado a función de flujo de control */
Flujo de Control (parámetro 1, parámetro 2, ..., parámetro n);

```

Figura 5-3: Código generado - Componentes CFP, DP

- **Generador de Archivos:** este módulo se encarga de generar nuevos archivos o ficheros (\*.java, \*.xml) tomando como referencia el código resultante del generador de código.
- **Resultado:** conjunto de clases java y archivos de configuración (\*.xml), los cuales son equivalentes al modelo gráfico realizado por el usuario. Todos estos archivos en conjunto representan la implementación de los servicios Telco 2.0 creados por el usuario de forma gráfica y están enmarcados dentro de los requerimientos de la especificación JAIN SLEE [17]. Cada servicio implementado es empaquetado y posteriormente desplegado en un servidor JAIN SLEE para probar su funcionamiento.

### 5.1.2.2 Diagrama de Despliegue

En la Figura 5-4 se observan los componentes del sistema, dentro de los cuales existen 3 servidores: un servidor Web, un servidor de base de datos y un SLEE; adicionalmente se encuentra un teléfono móvil que funciona como enrutador para el envío de mensajes de texto y, como clientes del sistema, un terminal SIP y un Browser, esto se explica a continuación:

- **SLEE:** es el componente principal de la arquitectura de negocio, puesto que gestiona la lógica, el acceso y ciclo de vida de los servicios.
- **Servidor Web:** servidor de aplicaciones web, donde se encuentran alojadas las aplicaciones web utilizadas por los servicios.
- **Cliente Web:** aplicaciones web, que le permiten al usuario, por medio de interfaces amigables e intuitivas, utilizar y acceder a las capacidades de cada uno de los servicios implementados en el SLEE.
- **Cliente SIP:** cliente SIP, el cual permite hacer uso de las funcionalidades de los servicios SIP (Sip Register, B2B, ClickToCall) mediante la creación de peticiones SIP.
- **Base de Datos:** servidor para bases de datos, utilizado para almacenar la información relacionada con cada servicio, creando una capa de acceso a datos para el sistema.
- **Enrutador de SMS:** dispositivo móvil utilizado como enrutador para el envío de mensajes de texto (SMS).

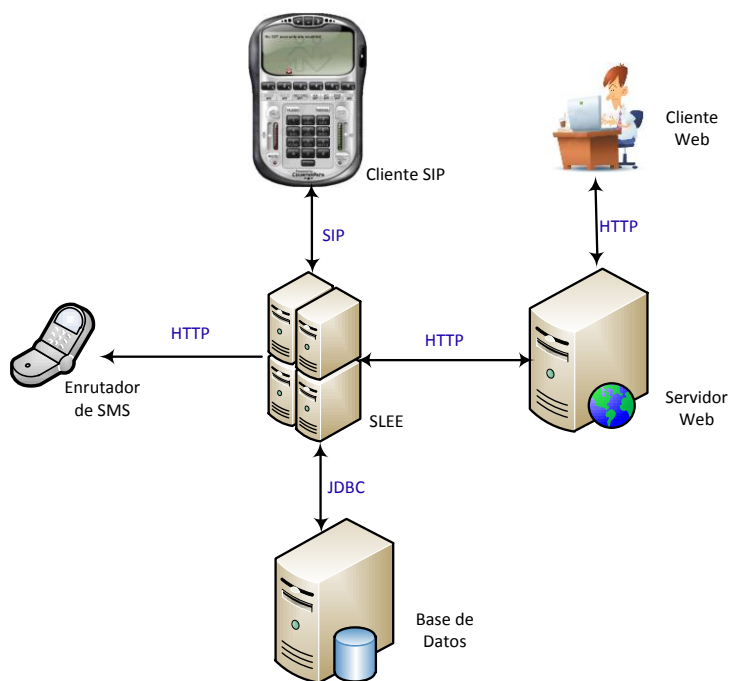


Figura 5-4: Diagrama de despliegue

### 5.1.2.3 Diagrama de Orquestación

En la Figura 5-5 se expone el diagrama de orquestación utilizado en el proceso de implementación y despliegue de los servicios Telco 2.0 desarrollados en el presente trabajo. El proceso de orquestación está enmarcado dentro de un modelo de componentes el cual permite reutilizar módulos y lograr las funcionalidades requeridas por un servicio, los principales componentes de este diagrama se explican a continuación.

- *SBB básicos*: Contienen la lógica de los servicios básicos, los cuales son módulos pre-implementados y son integrados dentro de la funcionalidad de un servicio convergente en particular.
- *SBB Orquestador*: este componente es el encargado de gestionar el orden de ejecución de los SBB o servicios básicos, dicha gestión se realiza a través del manejo de eventos (funcionalidad propia del SLEE), los cuales reaccionan ante acciones internas (eventos al interior del servicio) o externas (ej. eventos creados por un cliente) al proceso de orquestación, definiendo el comportamiento del orquestador con respecto a los demás componentes del sistema. Contiene la lógica de Orquestación de los servicios convergentes implementados y por medio de eventos específicos, el orquestador gestiona el orden de ejecución de los servicios básicos así como la reutilización de componentes.
- *Event Router*: modulo del SLEE encargado de enrutar los eventos internos y externos de la aplicación, cada servicio está compuesto por tres clases de eventos: evento http (onGET, onPOST); evento de entrada (onServiceStartEvent), evento de respuesta (onAnswerEvent), de esta forma este módulo se encarga de establecer la comunicación



entre el SBB orquestador y los SBB básicos, y de enrutar los eventos externos para el cumplimiento de una función específica.

- *Adaptadores de Recursos*: este módulo da la capacidad al SLEE de acceder a funcionalidades externas EJ: HttpRA (capacidades Http), SIPRA (capacidades SIP).

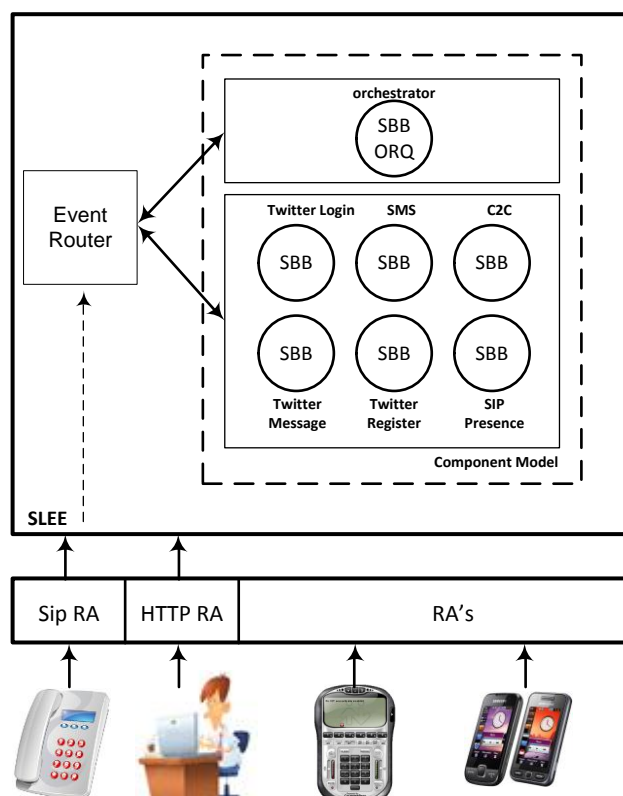


Figura 5-5: Diagrama de Orquestación

#### 5.1.2.4 Diagrama en paquetes, subsistemas y capas del sistema

La arquitectura lógica de la herramienta GT-4SC que se ilustra en la Figura 5-6 organiza el software utilizado en paquetes, subsistemas y capas. Esta se implementa en 3 capas: Aplicación, Mediación y Fundamentación; en donde son presentadas las interacciones entre los paquetes que componen cada capa; a continuación se describe cada una de ellas.

**Capa de Aplicación:** gestiona los paquetes que implementan las funcionalidades del prototipo, está compuesta por los siguientes paquetes:

- *JSPs*: módulo utilizado para crear las interfaces de las aplicaciones web clientes.
- *Servlets*: contiene los Servlets encargados de la interfaz de administración de las aplicaciones web clientes, y captura decisiones y eventos generados por el usuario.
- *GT-4SC*: contiene una extensión de Eclipse que permite componer servicios convergentes de forma gráfica utilizando bloques que representan servicios básicos previamente implementados y patrones de flujo de control.

- *Servicios Básicos*: este paquete contiene los servicios básicos previamente implementados, los cuales sirven como base para componer nuevos servicios.
- *Orquestador*: contiene al servicio que coordina las interacciones entre servicios básicos, así como también las interacciones entre eventos externos o internos y servicios.
- *Cliente Sip*: software utilizado para originar eventos y peticiones SIP las cuales serán capturadas por el servidor. Para este prototipo se seleccionaron tres versiones de clientes SIP: Twinkle, Xlite y VMobile.

**Capa de mediación:** contiene todas las interfaces de aplicación de programa (APIs) utilizadas por el prototipo, está compuesta por los paquetes:

- *Apache Tomcat*: implementa los servlets y las Java Server Pages (JSP) especificados por Oracle Corporation, proporcionando un entorno para ejecutar código Java en cooperación con un servidor Web. Éste adiciona herramientas para gestión y configuración pero también puede ser configurado a través de archivos de configuración en formato XML. Para este prototipo se ha seleccionado un servidor de aplicaciones web Tomcat V7.0.22
- *OAuth*: contiene la implementación de un protocolo abierto de autenticación que permite a los usuarios aprobar, en su nombre, a una aplicación Web sin exponer su contraseña. Para el caso de Twitter, una aplicación externa puede solicitar al usuario que ingrese su nombre de usuario y contraseña de Twitter para ingresar a esta [76]. Esta forma de acceso a aplicaciones externas a Twitter con los datos, se maneja mediante access tokens los cuales pueden expirar en un tiempo determinado según las políticas manejadas. Las especificaciones de este protocolo de acceso se encuentran en [77] [78].
- *Twitter4J*: contiene una librería no oficial para la integración del API de Twitter con cualquier aplicación JAVA. Esta librería es fácilmente agregada al classpath de la aplicación que se desea desarrollar y funciona con una versión igual o superior de JVM 1.4.2 [79].
- *Draw2D*: permite mostrar gráficos utilizando SWT Canvas<sup>17</sup>;
- *GEF*: (Graphical Editor Framework) es un framework que tiene la capacidad de crear vistas y ediciones gráficas para eclipse. Dentro de estas capacidades, se incluye la creación de diagramas de actividades, máquinas de estado, diagramas de clases, entre otros.
- *JET*: este módulo contiene el framework JET (Java Emitter Template) utilizado para la generación de código. Es usado para la inserción de código de modelos gráficos abstractos, por ejemplo, si un usuario realiza un diagrama de clases utilizando GEF, para que la maquina entienda lo que este representa, es necesario generar código que la maquina pueda procesar para entender dicho diagrama; lo anterior es precisamente la función de JET [66].
- *Mobicents JainSlee*: es la primera y única plataforma de libre distribución certificada para JainSlee V1.1 la cual integra un servidor de aplicaciones basado en eventos, altamente escalable, y con soporte a fallos. La información complementaria a esta plataforma se explica en la Sección 2.1.5 y el código fuente puede ser descargado de [80].

---

<sup>17</sup> SWT: Standard Widget Toolkit, utilizado por Eclipse para el acceso a las librerías nativas graficas en Java.

- *EclipSlee*: contiene el Plug-in Mobicents *versión 1.2.6*: el cual se integra con Eclipse para soportar y ayudar con la creación de componentes JainSlee como: eventos, bloques constructores de servicio (SBB), especificaciones de perfil, descriptores XML, adaptadores de recursos y unidades desplegadas. Puede ser descargado de [81].
- *JBoss*: es un servidor de aplicaciones J2EE de código abierto implementado en Java, para el desarrollo y despliegue de aplicaciones empresariales como servicios, portales Web etc. Dentro de sus principales características están: el soporte de transporte mediante HTTP, Servlets, Bisockets, RMI; descubrimiento automático a través de JNDI y Multicast; llamadas asíncronas punto a punto o a través de un servidor; compresión de datos; entre muchas otras. La información de instalación y despliegue de este servidor se encuentra en [82]. La versión seleccionada para este prototipo es *JBoss 2.5.0*.
- *JAIN SLEE*: gestiona la lógica, el acceso y ciclo de vida de los servicios. Para este prototipo se seleccionó la distribución JAIN SLEE Mobicents V1.1, el cual es un SLEE robusto de libre distribución.
- *JDBC*: es el conector para base de datos de Java. Para este prototipo se utilizó la librería postgresql-8.4-703.

**Capa de Fundamentación:** Incluye el software básico que permite el rendimiento del prototipo, está compuesta por los siguientes paquetes:

- *Postgres SQL*: es un motor de administración de bases de datos relacionales. La versión seleccionada para este prototipo fue Postgres 8.4 y el gestor de base de datos Postgres pgAdmin versión 3.
- *JDK*: corresponde al entorno de desarrollo de aplicaciones Java. La versión seleccionada para este prototipo fue JDK 6
- *Eclipse IDE*: contiene el IDE (Entorno de Desarrollo Integrado) Eclipse Indigo 3,7 el cual permite el desarrollo de aplicaciones Web; integración con entornos de creación y ejecución de servicios como JAIN SLEE; creación de modelos gráficos utilizando GEF, entre muchas otras funcionalidades adicionales. Esta versión puede ser descargada en [75]. La importancia de este entorno es el soporte que la comunidad Eclipse le brinda, ya que facilita que nuevos usuarios desarrollen sus primeros productos software, con una gran cantidad de libre información.
- *Ubuntu 10.04*: sistema operativo que soporta el prototipo.

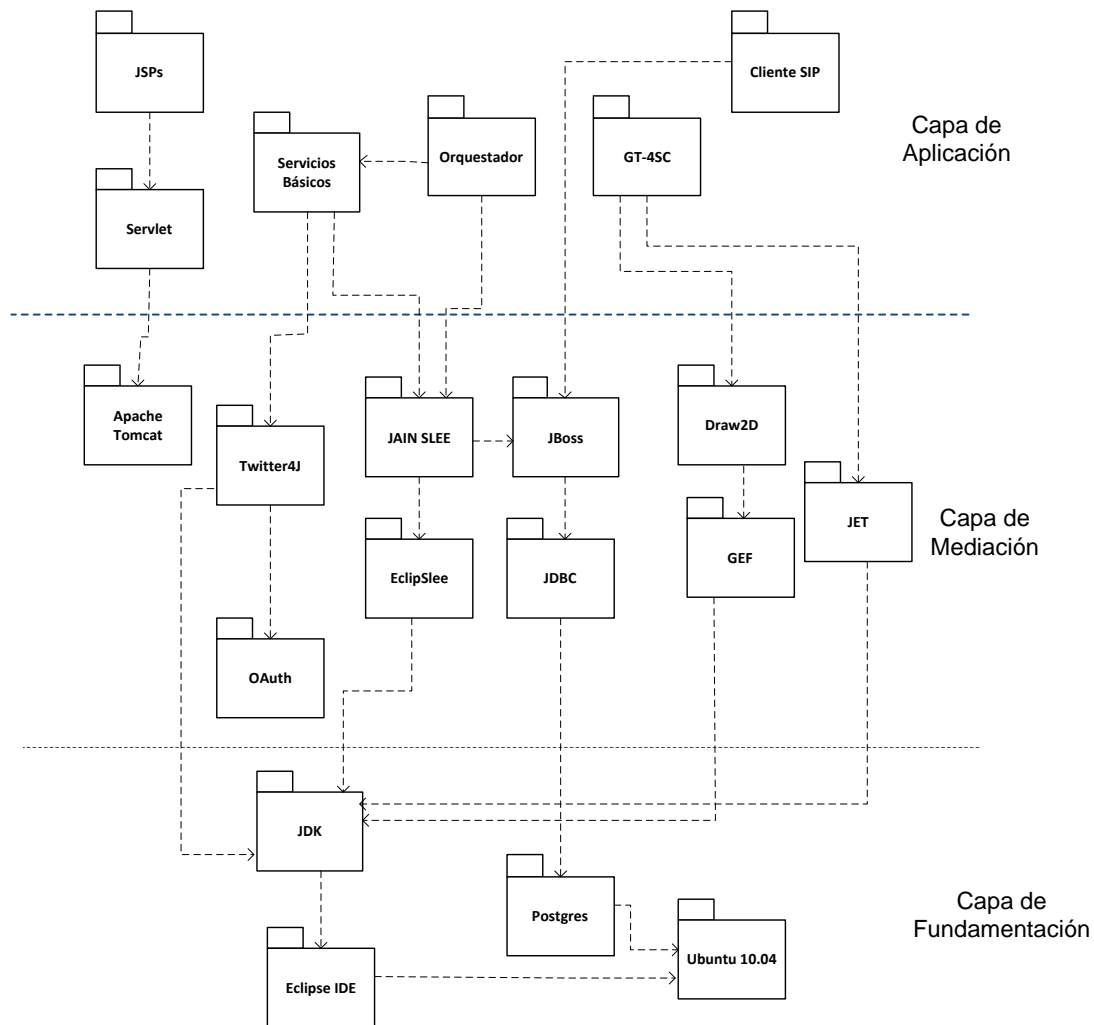


Figura 5-6: Diagrama en paquetes, subsistemas y capas del sistemas

### 5.1.3 Consideraciones generales del prototipo

La herramienta GT-4SC permite la creación de nuevos servicios mediante la combinación de servicios básicos previamente implementados; estos servicios son desplegados y ejecutados en un servidor JAIN SLEE, el cual cumple con los requerimientos de los servicios Telco 2.0.

Adicionalmente, la herramienta GT-4SC es escalable en términos de servicios y patrones de flujo de ejecución, ya que es posible agregar nuevos servicios y patrones siguiendo un conjunto de reglas específicas de implementación, las cuales se describen a continuación.

Si se desea incluir un nuevo servicio se deben seguir los siguientes pasos:

- Implementar el nuevo servicio con dos eventos propios, uno de entrada y otro de respuesta (StartServiceEvent, AnswerServiceEvent) y un evento de recepción http (POST o GET), de esta forma el servicio tendrá un canal de comunicación con el orquestador y con clientes externos si es necesario.

- Crear la unidad desplegable del servicio y agregarla a la herramienta GT-4SC de tal forma que la herramienta pueda hacer uso del nuevo servicio cuando lo necesite.
- Referenciar el nuevo servicio en el módulo de generación de código automática de la herramienta GT-4SC con el fin generar el código necesario (referencias, importación de eventos del nuevo servicio, etc.) en el momento en que el nuevo servicio sea agregado al área de trabajo.

Si se desea ingresar un nuevo patrón, se debe seguir los siguientes pasos

- Crear el código de implementación del nuevo patrón, especificando los parámetros necesarios para el funcionamiento de este.
- Referenciar el nuevo patrón en el módulo de generación de código automática de la herramienta GT-4SC, con el fin generar el código necesario (referencias, especificación de parámetros, implementación del patrón, etc.) en el momento en que el nuevo patrón sea agregado al área de trabajo.

Para el presente trabajo de grado, se implementaron tres servicios Telco 2.0 (twittSMS, CallMessage, TwitterCalls) utilizando la herramienta GT-4SC; estos servicios sirvieron como prueba funcional de la herramienta y permitieron demostrar las ventajas de la composición de servicios convergentes por medio de patrones de flujo de ejecución.

#### **5.1.4 Interfaces gráficas de usuario del sistema y Descripción de Servicios Implementados**

En esta sección se exponen las interfaces gráficas de tres servicios de prueba creados con la herramienta GT-4SC, de igual forma se describe su funcionamiento y algunas consideraciones para cada servicio.

##### **5.1.4.1 Servicio TwittSMS**

*Descripción del Servicio:* este servicio permite, enviar un SMS o un mensaje directo de Twitter a un usuario de destino.

*Consideraciones:* tanto el usuario emisor como el receptor, deben estar registrados en la base de datos de la aplicación, adicionalmente el usuario receptor debe seguir en Twitter al usuario emisor, debido a que solo de esta forma los mensajes directos de Twitter son autorizados por la red social.

Las interfaces de este servicio se muestran en el Anexo D.

##### **5.1.4.2 Servicio CallMessage**

*Descripción del Servicio:* este servicio permite a un usuario realizar una llamada SIP y comunicarse con un usuario de destino. Si la llamada no es exitosa, el servicio envía automáticamente un mensaje directo de Twitter con la información de llamada, el cual será recibido por el usuario de destino.

*Consideraciones:* tanto el usuario que origina la llamada como el de destino, deben estar registrados en la base de datos de la aplicación, adicionalmente los dos usuarios deben estar registrados en un terminal SIP (softphone), con el fin de recibir los eventos SIP originados desde la interfaz web; finalmente, el usuario de destino debe seguir en Twitter al usuario que origina la llamada, debido a que solo de esta forma los mensajes directos de Twitter son autorizados por la red social.

Las interfaces de este servicio se muestran en el Anexo D.

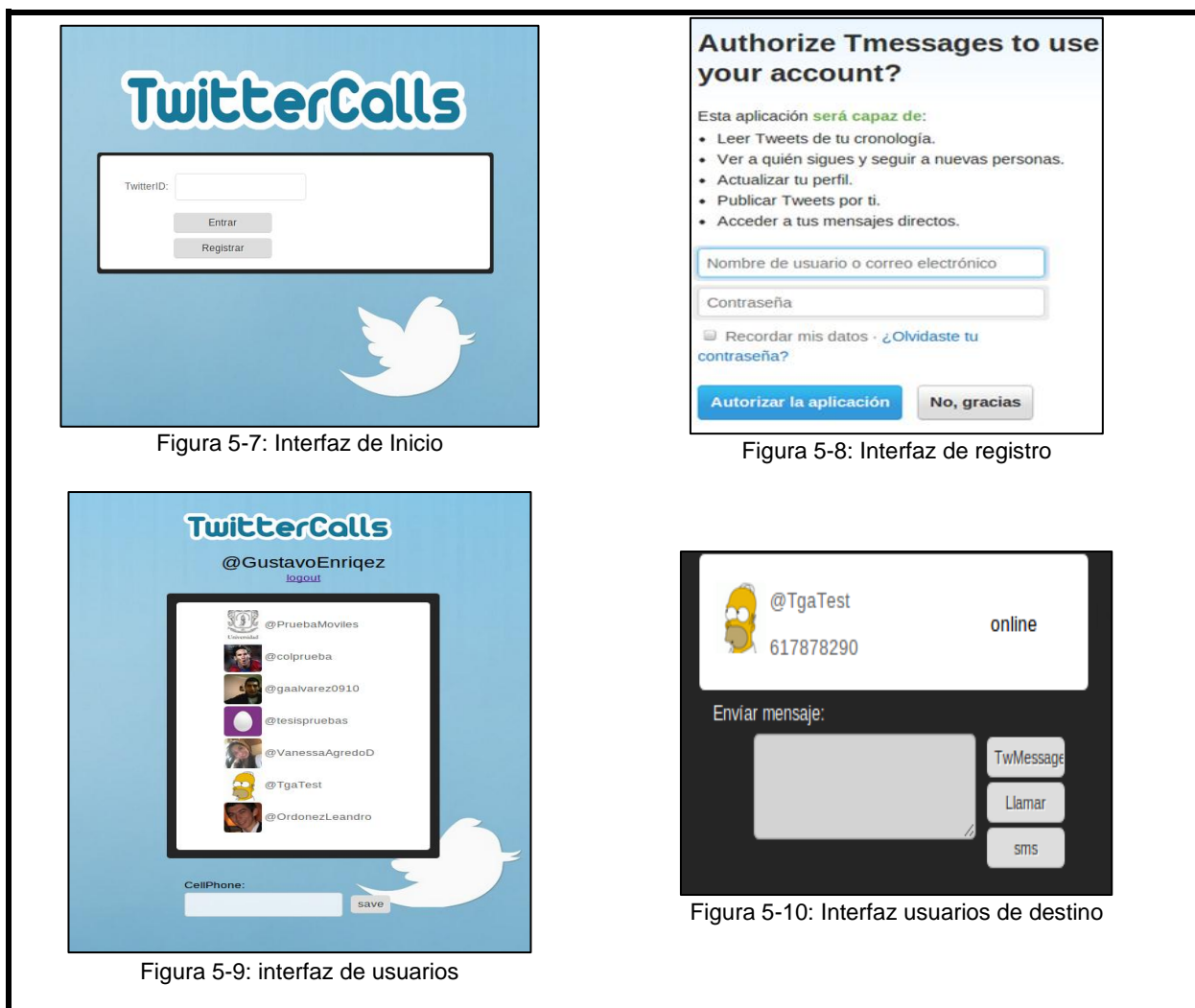
### 5.1.4.3 Servicio TwitterCalls

*Descripción del Servicio:* este servicio permite a un usuario registrarse al servicio TwitterCalls y en caso de haberse registrado anteriormente ingresar a esta utilizando su Twitter Screen ID. Una vez haya ingresado a la aplicación, el usuario podrá seleccionar a un usuario de destino y enviarle un SMS, un mensaje directo de Twitter o llamarlo. Si el servicio de llamada SIP fue el escogido y la comunicación no es exitosa, el servicio TwitterCalls enviará un SMS y un mensaje directo de Twitter al usuario de destino con la información de la llamada.

*Consideraciones:* ninguna.

*Interfaces de usuario:*

- Interfaz de inicio: la interfaz de inicio (Figura 5-7), permite al usuario registrarse a la aplicación; si el usuario ya se ha registrado previamente, podrá ingresar haciendo uso de su Twitter Screen IDn (ej: @usuario).
- Interfaz de registro: el registro a la aplicación se realiza directamente con la red social Twitter (Figura 5-8), por lo cual el servicio TwitterCalls no tendrá acceso a las contraseñas de los usuarios. Una vez registrado, el usuario ingresará a la aplicación donde se encontrará con la *interfaz de usuarios*.
- Interfaz de usuarios: en la interfaz que se muestra en la Figura 5-9 el usuario encuentra una lista de seguidores que están registrados en la aplicación y haciendo click sobre la foto de cada uno de ellos podrá acceder a varios servicios que ofrece TwitterCalls.
- Interfaz usuario de destino: la Figura 5-10 es un ejemplo de la interfaz de usuarios de destino, donde se muestran los datos de éste (foto, twitter screen ID, twitter ID, estado SIP) y ofrece tres servicios: enviar un mensaje directo de Twitter, enviar un SMS o realizar una llamada SIP.



En el Anexo D se describe un ejemplo de funcionamiento de este servicio.

## 5.2 Experimentación y pruebas

En esta Sección se presenta el plan de pruebas, el cual se realizó siguiendo la metodología que utiliza López *et al.* en [83], la cual se divide en dos fases principales: *planeación* y *ejecución del experimento* (Sección 5.2.1). En la primera de estas se define el objetivo, hipótesis, selección de sujetos y objetos experimentales; y en la segunda, se elabora el plan para la ejecución de las pruebas con usuarios.

Finalmente, en la Sección 5.2.2 se presentan los resultados obtenidos, los cuales son analizados mediante estadística descriptiva e inferencial.

## 5.2.1 Plan de pruebas

El plan de pruebas que se presenta a continuación se divide en dos fases, la primera es la fase de planeación del experimento donde se describe el objetivo de las pruebas, la formulación de la hipótesis, selección de variables y la selección de sujetos y objetos experimentales. La segunda fase corresponde al plan de ejecución de pruebas, donde se explica cómo es el proceso de recolección de datos. Las anteriores fases se explican a continuación.

### 5.2.1.1 Planeación del experimento

**Definición Objetivo:** medir el tiempo de composición de tres servicios convergentes llamados: “TwitterCalls”, “TwitterSMS” y “CallMessage” haciendo uso de la herramienta **GT-4SC** y contrastar estos resultados con el tiempo empleado para la composición de los mismos en código fuente sin el uso de patrones de flujo de ejecución.

#### Formulación de Hipótesis:

- H0 (Hipótesis Nula): la composición de servicios Telco 2.0 mediante patrones de flujo de ejecución **NO reduce** el tiempo de composición, en comparación al tiempo empleado cuando se realiza sin patrones.
- Ha (Hipótesis alternativa): la composición de servicios Telco 2.0 mediante patrones de flujo de ejecución **reduce** el tiempo de composición, en comparación al tiempo empleado cuando se realiza sin patrones.

#### Selección de variables:

- **Independiente:** Patrones de flujo de ejecución.
- **Dependiente:** Tiempo de composición del servicio.

**Selección de sujetos:** El experimento se desarrolla con 15 estudiantes de la Universidad del Cauca (estudiantes del programa de Ingeniería Electrónica y Telecomunicaciones de último semestre y estudiantes del programa de maestría en Ingeniería Telemática de primer año) los cuales tienen conocimientos de programación orientada a objetos y además cuentan con experiencia en el desarrollo de servicios en JAIN SLEE.

**Objetos experimentales:** se desea conocer el tiempo de composición de tres servicios convergentes llamados “TWITTERCALLS” (Twitter + SMS + Call), “TWITTERSMS” (Twitter + SMS) y “CALLMESSAGE” (Call + TwitterMessage), utilizando dos formas de programación: la primera utiliza la herramienta llamada: **GT4SC (Graphical Tool-patterns-based for Service Creation)** la cual funciona sobre el entorno ECLIPSE y utiliza un Plug-in de JAIN SLEE llamado ECLIPSLEE; y la segunda, se diferencia de la primera, ya que los sujetos de prueba no disponen de **GT4SC** para la creación de los servicios.

Se tuvieron en cuenta unas condiciones generales para todos los experimentos, las cuales se describen a continuación:



- Se escogieron 5 sujetos al azar para cada prueba (de 15), los cuales han sido invitados a ingresar a un laboratorio, en el cual se les puso el objetivo de crear los tres servicios ya nombrados. Antes del inicio de la prueba se les informo en que consiste el servicio que deben desarrollar, sin embargo, es hasta que ingresan al laboratorio, que fueron notificados de las herramientas software que iban a utilizar para desarrollar.
- Se hizo entrega de una guía de desarrollo, la cual describe cómo utilizar cada una de estas herramientas, para que los servicios puedan ser creados, ejecutados y desplegados.
- Cuando el sujeto de prueba inició la creación del servicio, el tiempo comenzó a correr, sin que éste haya sido notificado (para evitar presiones y ansiedad).
- El tiempo se detuvo cuando el sujeto: compuso el servicio y creo la unidad desplegable del mismo. Finalmente, con el reloj detenido, se realizó una prueba de funcionamiento de todo el servicio. Al momento de presentarse algún error en cualquiera de estas pruebas, el cronometro volvió al correr hasta que los problemas fueron solucionados.

### 5.2.1.2 Plan de ejecución de pruebas

Una vez explicadas las condiciones generales, en esta sección se explica, cómo fueron ejecutadas las pruebas.

El experimento se dividió en dos pruebas principales: **T1** y **T2**; a continuación se explican cada una de ellas

- **T1:** se formaron dos grupos de 5 personas al azar; cada grupo se etiquetó como: “Experimentación A” y “Experimentación B”, los cuales crearon los servicios nombrados anteriormente. El primer grupo utilizo las siguientes herramientas software: Herramienta grafica **GT4SCE**, Eclipse IDE, EclipSLEE, Mobicents,, Servicios empaquetados de: Registro, Click to Dial, SMS, TwitterAP; por otra parte, el segundo grupo utilizo las mismas herramientas software, menos la herramienta grafica **GT4SCE**. A cada sujeto de prueba, se le tomo el tiempo empleado en la creación de servicio y su correspondiente unidad desplegable. La imagen que ilustra la metodología de la prueba **T1** se muestra en la Figura 5-11.
- **T2:** la metodología de esta prueba es similar a la utilizada para **T1**, la diferencia está en que solo se escogieron cinco sujetos de pruebas los cuales realizaron las dos experiencias, tanto A como B, con el fin de medir el desempeño de cada uno de ellos, utilizando ambas formas de programación. La imagen que ilustra la metodología de esta prueba se muestra en la Figura 5-12.

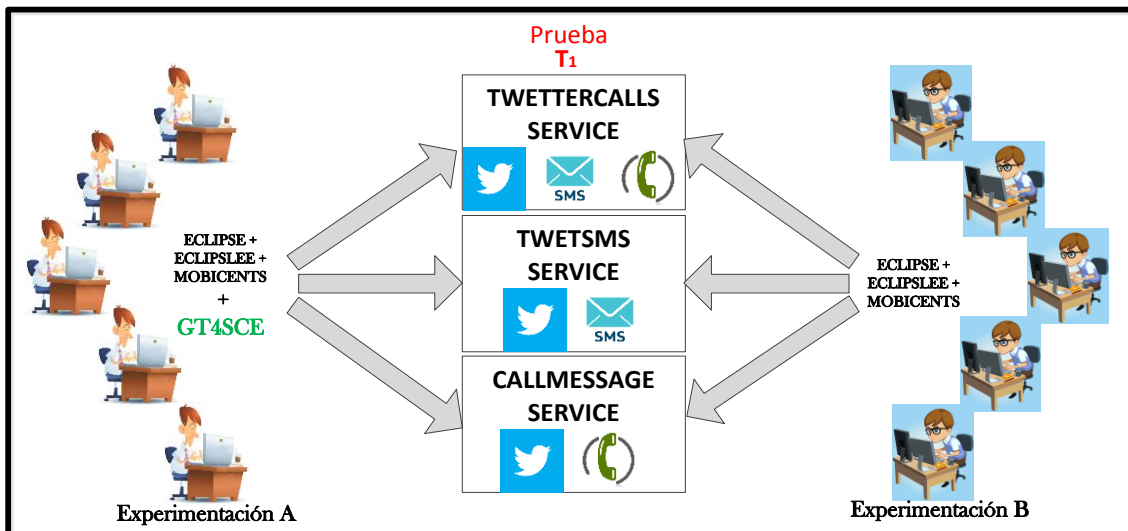


Figura 5-11: Ilustración de la Prueba T1

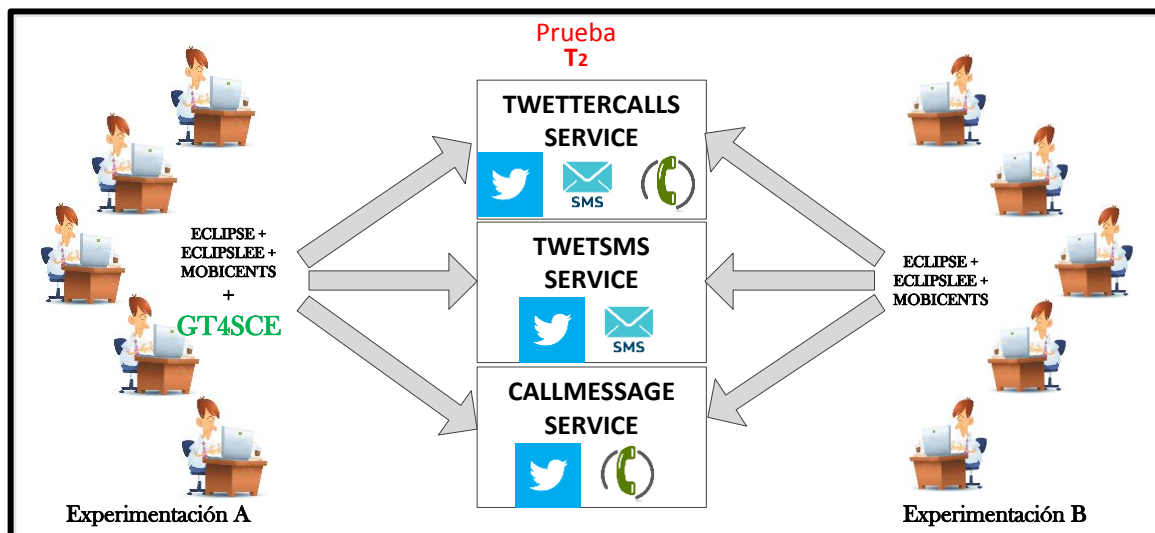


Figura 5-12: Ilustración de la Prueba T2

## 5.2.2 Resultados obtenidos

Para analizar los resultados obtenidos se utilizó estadística descriptiva e inferencial. La primera representa y resume los datos de las variables de una muestra determinada; y la segunda se basa en el contraste de la hipótesis, la cual busca cuantificar si las tendencias/asociaciones observadas en el análisis descriptivo son ciertas o si son el resultado de fluctuaciones aleatorias causadas por la variabilidad propia de una muestra [84].

A continuación en la Sección 5.2.2.1 se presenta el análisis descriptivo y en la Sección 5.2.2.2 el análisis inferencial.

### 5.2.2.1 Resultados obtenidos descritos mediante estadística Descriptiva

Los resultados obtenidos en la Prueba **T1** se estructuran según la Tabla 5-1

Tabla 5-1: Estructura de la representación Gráfica de T1

TwitterCalls Experimentación A (Figura 5-13)	TwitterCalls Experimentación B (Figura 5-14)
TwitterSMS Experimentación A (Figura 5-15)	TwitterSMS Experimentación B (Figura 5-16)
CallMessage Experimentación A (Figura 5-17)	CallMessage Experimentación B Figura 5-18

Los resultados obtenidos en la Prueba **T2** se estructuran según la Tabla 5-2.

Tabla 5-2: Estructura de la representación Gráfica de T2

TwitterCallsExperimentación A y B (Figura 5-19)
TwitterSMS Experimentación A y B (Figura 5-20)
CallMessage Experimentación A y B (Figura 5-21)

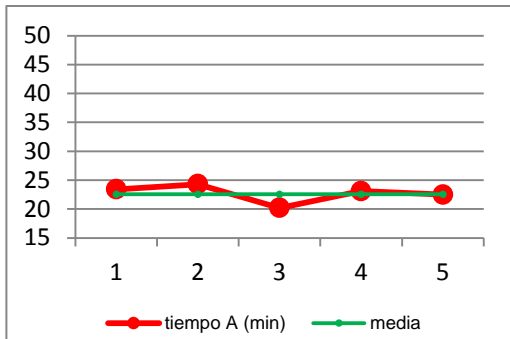


Figura 5-13: TwitterCalls Experimentación A – T1

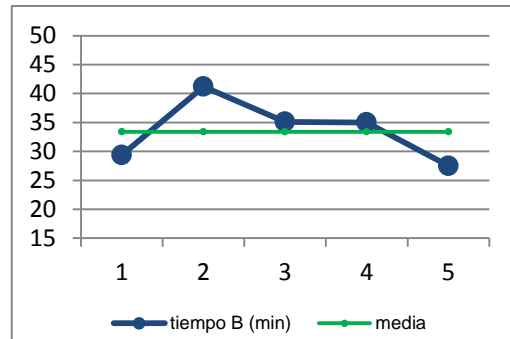


Figura 5-14: TwitterCalls Experimentación B – T1

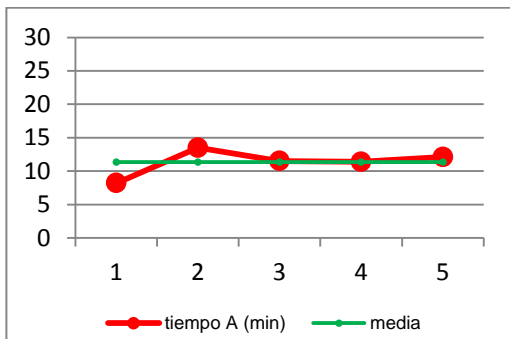


Figura 5-15: TwitterSMS Experimentación A – T1

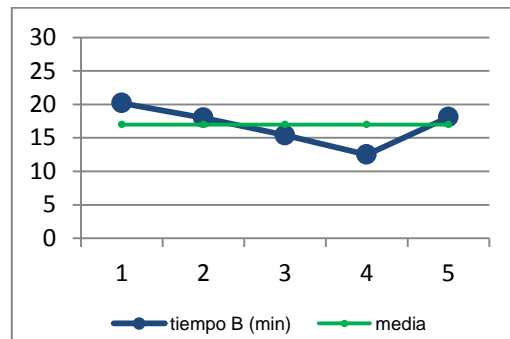


Figura 5-16: TwitterSMS Experimentación B – T1

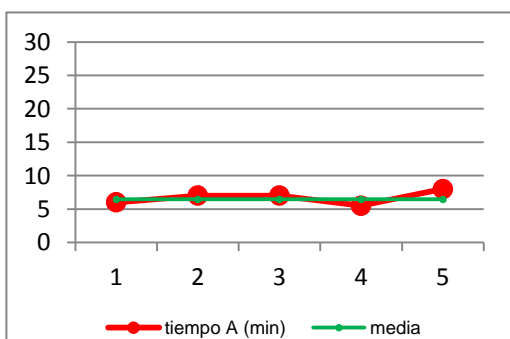


Figura 5-17: CallMessage Experimentación A – T1

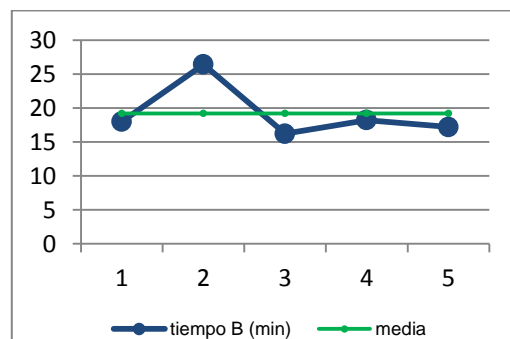


Figura 5-18: CallMessage Experimentación B – T1

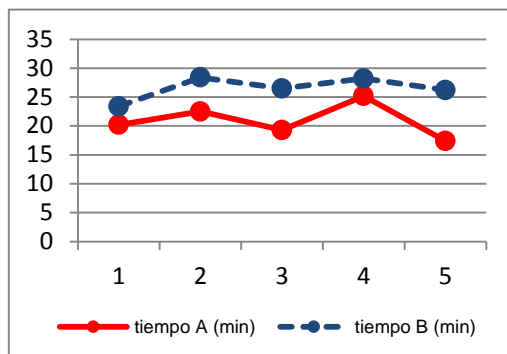


Figura 5-19: TwitterCalls Experimentación A y B - T2

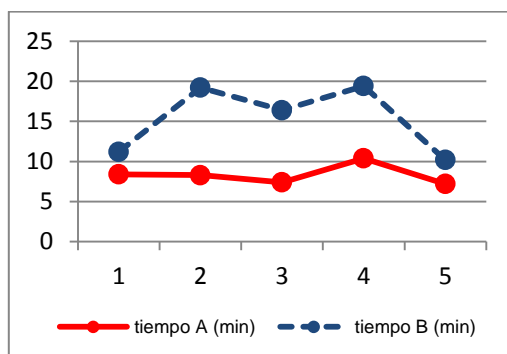


Figura 5-20: TwitterSMS Experimentación A y B - T2

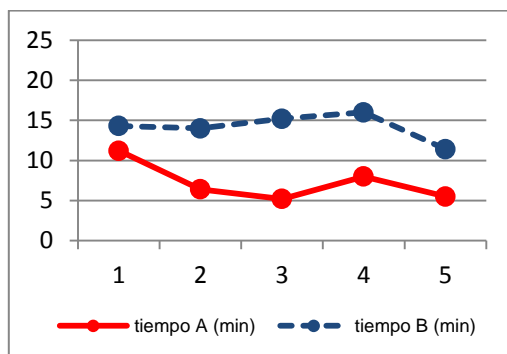


Figura 5-21: CallMessage Experimentación A y B - T2

Las anteriores graficas muestran claramente que el tiempo empleado por los usuarios en la Experimentación A para la prueba T1 como T2, es considerablemente menor al tiempo empleado por los usuarios en la Experimentación B.

Lo anterior demuestra que la composición de servicios convergentes mediante patrones de flujo de ejecución, bajo las condiciones del experimento, reduce el tiempo en comparación al proceso de composición realizado sin patrones; sin embargo es necesario aplicar una prueba

inferencial que permita rechazar o aceptar la hipótesis planteada; lo anterior se describe en la Sección 5.2.2.2.

### 5.2.2.2 Resultados obtenidos descritos mediante estadística Inferencial

Para analizar los resultados obtenidos en las pruebas **T1** y **T2**, se utilizó la prueba t-Student, la cual permite inferir a partir de pequeñas muestras, un rango de aceptación o rechazo de la hipótesis, con un valor determinado de aceptación [85].

Esta prueba se utiliza cuando una muestra es pequeña<sup>18</sup>, sin embargo se necesitan satisfacer dos condiciones para poder aplicarla, la primera de estas explica que la muestra debe ser homogénea, para tratar de reducir al máximo el error; para el caso del experimento planteado, se han escogido sujetos de prueba los cuales estudian o estudiaron el mismo programa universitario y tienen conocimientos similares de programación; y la segunda de estas es la similitud de varianzas; sin embargo esta última condición solo se puede determinar una vez el experimento haya sido aplicado.

La prueba t-Student, presenta variaciones que dependen de las condiciones del experimento; la primera se utiliza para realizar inferencias de la media de una población, con respecto a una muestra; la segunda se utiliza para realizar inferencias con muestras pequeñas, acerca de la diferencia de dos medias; y la tercera se utiliza cuando las muestras se denominan “pareadas” o dependientes.

La ecuación general para el valor  $t$ , de la prueba t-Student viene dado por la ecuación (2).

$$t = \frac{y - \mu}{s/\sqrt{n}} \quad (2)$$

$\therefore y$  = media;  $\mu$  es el valor de la hipótesis nula,  $s$  es la varianza y  $n$  es el número de muestras.

La fórmula de la varianza viene dada por la expresión (3).

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - y)^2} \quad (3)$$

$\therefore n$  = Total de Sujetos de experimentación.

$y_i$  = Valor asociado a cada sujeto de prueba  $i \therefore i = \{1, 2 \dots 5\}$

$y$  = Media estadística.

Como se observa en las ecuaciones (2) y (3) el valor de  $t$ , depende de los  $n - 1$  sujetos de prueba, a lo que también se le conoce como  $n - 1$  grados de libertad, este valor es muy importante ya que la prueba  $t$  varía con relación a los grados de libertad, como se observa en la tabla que se encuentra en el Anexo F, además los diferentes grados de libertad, tienen asociado un determinado  $\alpha$ , el cual indica el porcentaje de confiabilidad de la prueba, por

<sup>18</sup> Muestra pequeña: Se considera una muestra pequeña cuando es inferior a 30.

ejemplo: para  $\alpha = 0.05$ , existe un 95% de confiabilidad de que la hipótesis se acepte o rechace.

Por otra parte, la prueba se evalúa de la siguiente manera: si  $t > t_\alpha$  la hipótesis nula se rechaza y la hipótesis alternativa se acepta, de lo contrario, la hipótesis alternativa se rechaza y la nula se acepta.

Como se dijo al comienzo de ésta Sección, existen tres formas de realizar la prueba t-Student; debido a las condiciones que se plantearon para los experimentos de este trabajo, se utiliza la prueba t-Student para dos muestras independientes (se aplica para **T1**) y t-Student para muestras pareadas (se aplica para **T2**). A continuación se explican las dos pruebas.

### t-Student para muestras independientes

Las expresiones (2) y (3) para determinar el valor de  $t$  y la varianza  $s$ , solo consideran una única muestra, por esta razón es necesario modificar las expresiones para dos muestras como se describe en las ecuaciones (4) (5).

$$s = \sqrt{\frac{1}{n_1 + n_2 - 2} \left( \sum_{i=1}^n (y_i - y_1)^2 \right) + \left( \sum_{i=1}^n (y_i - y_2)^2 \right)} \quad (4)$$

$$t = \frac{(y_1 - y_2) - D_0}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (5)$$

Por otra parte las expresiones para las hipótesis, se muestran a continuación.

$$\begin{aligned} \text{Hipótesis nula } H_o &= (\mu_1 - \mu_2) = D_0 \\ \text{Hipótesis alternativa } H_a &= (\mu_1 - \mu_2) \neq D_0 \end{aligned}$$

El procedimiento de esta prueba consiste en calcular el valor de la varianza  $s$  con la formula (4) teniendo en cuenta los resultados obtenidos en las Experiencias A y B para **T1**. A continuación, se calcula el valor de  $t$  con la formula (5), y finalmente se compara el resultado obtenido con los valores de la distribución t-Student para un determinado  $\alpha$ .

La Tabla 5-3 muestra los resultados obtenidos. En la columna *Servicios* se encuentran los servicios que los usuarios crearon; la columna *Tiempo A* contiene el tiempo empleado por cada usuario para la composición de los servicios utilizando la herramienta grafica basada en patrones **GT4-SC**; la columna *Tiempo B* contiene el tiempo empleado por cada usuario para la composición de los servicios a nivel de código; la columna *Varianza* muestra el cálculo de la varianza para cada experimento, el cual se hizo utilizando la formula (4); la columna  $t$  muestra el cálculo del valor de  $t$  mediante la fórmula (5); finalmente, la columna  $t_{\alpha=0.025}$  muestra el valor definido por la distribución t-Student para un porcentaje de rechazo de la hipótesis nula  $H_o$ , para un 97,5% (Ver Anexo F).

Tabla 5-3: Análisis inferencial de los resultados obtenidos en la Prueba T1

Servicio	Tiempo A	Tiempo B	Varianza	$t$	$t_{\alpha=0.025}$
TwitterCalls	23,4	29,4	3,57	4,35	2,3060
	24,3	41,2			
	20,2	35,1			
	23,1	35			
	22,5	27,5			
TwitterSMS	8,2	20,2	3,02	3,46	
	13,5	18			
	11,5	15,4			
	11,4	12,5			
	12,1	18,1			
CallMessage	6	18	3,01	6,63	
	7	26,4			
	7	16,2			
	5,5	18,2			
	8	17,2			

Teniendo en cuenta los resultados obtenidos, el paso final consiste en evaluar estos resultados con la expresión  $t > t_{\alpha}$ .

- Servicio TwitterCalls:  $t = 4,35$  ;  $t_{\alpha=0.025} = 2,3060$ , entonces  $t > t_{\alpha}$ , por lo cual la Hipótesis nula  $H_0$  se rechaza, y se acepta la Hipótesis alternativa  $H_a$  con un porcentaje de 97,5% de probabilidad, de que si el experimento se vuelve a realizar bajo las mismas condiciones planteadas, la composición del servicio convergente TwitterCalls mediante patrones de flujo de ejecución reduce el tiempo de composición en comparación al tiempo empleado cuando no se utilizan patrones.
- Servicio TwitterSMS:  $t = 3,46$  ;  $t_{\alpha=0.025} = 2,3060$ , entonces  $t > t_{\alpha}$ , por lo cual la Hipótesis nula  $H_0$  se rechaza, y se acepta la Hipótesis alternativa  $H_a$  con un porcentaje de 97,5% de probabilidad, de que si el experimento se vuelve a realizar bajo las mismas condiciones planteadas, la composición del servicio convergente TwitterSMS mediante patrones de flujo de ejecución reduce el tiempo de composición en comparación al tiempo empleado cuando no se utilizan patrones.
- Servicio CallMessage:  $t = 6,63$  ;  $t_{\alpha=0.025} = 2,3060$ , entonces  $t > t_{\alpha}$ , por lo cual la Hipótesis nula  $H_0$  se rechaza, y se acepta la Hipótesis alternativa  $H_a$  con un porcentaje de 97,5% de probabilidad, de que si el experimento se vuelve a realizar bajo las mismas condiciones planteadas, la composición del servicio convergente CallMessage mediante patrones de flujo de ejecución reduce el tiempo de composición en comparación al tiempo empleado cuando no se utilizan patrones.

### t-Student para muestras pareadas

Esta prueba permite hacer inferencias cuando se hacen dos experimentos con los mismos sujetos experimentales, como es el caso de la prueba **T2**. Las expresiones para obtener el valor de  $t$  son similares a (2) y (3) sin embargo no se obtiene el valor media y a partir de los tiempos de composición de cada usuario, se obtiene a partir del promedio de la diferencia en



tiempo, lo cual se rotula como  $d$ ; para el caso de este experimento, es el promedio de la diferencia entre el tiempo A y B. Por otra parte, la varianza  $S$ , se calcula teniendo en cuenta la diferencia entre el tiempo A y B, y la media  $d$ , debido a este cambio, la varianza se rotula  $S_d$  y está dada por la formula (6).

Por lo anterior la formula (2) para el cálculo de  $t$ , cambia como se muestra en la ecuación (7).

$$s_d = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - d)^2} \quad (6)$$

$$t = \frac{d - \mu}{s_d / \sqrt{n}} \quad (7)$$

El proceso para el cálculo de  $t$ , es exactamente el mismo al utilizado en la prueba **T1**, y los resultados se muestran en la Tabla 5-4. En la columna *Servicios* se encuentran los servicios que los usuarios crearon; la columna *Tiempo A* contiene el tiempo empleado por cada usuario para la composición de los servicios utilizando la herramienta grafica basada en patrones **GT4-SC**; la columna *Tiempo B* contiene el tiempo empleado por cada usuario para la composición de los servicios a nivel de código; la columna  $d$  muestra el cálculo de la media de las diferencias en tiempo de A y B; la columna  $S_d$  corresponde al cálculo de la varianza utilizando la formula (6); la columna  $t$  muestra el cálculo del valor de  $t$  mediante la fórmula (7); y finalmente, la columna  $t_{\alpha=0.025}$  muestra el valor definido por la distribución t-Student para un porcentaje de rechazo de la hipótesis nula  $H_0$ , para un 97,5% (Ver Anexo F).

Tabla 5-4: Análisis inferencial de los resultados obtenidos en la Prueba T2

Servicio	Tiempo A	Tiempo B	$d$	$s_d$	$t$	$t_{\alpha=0.025}$
<b>TwitterCalls</b>	20,4	23,4	5,34	2,52	<b>4,98</b>	2,7765
	22,5	28,4				
	19,3	26,5				
	25,2	28,2				
	17,4	26,2				
<b>TwitterSMS</b>	8,4	11,2	6,54	3,46	<b>4,11</b>	
	8,3	19,2				
	7,4	16,4				
	10,4	19,4				
	7,2	10,2				
<b>CallMessage</b>	11,2	14,3	6,52	2,58	<b>5,98</b>	
	6,4	14				
	5,2	15,2				
	8	16				
	5,5	11,4				

Una vez obtenidos los resultados, el paso final consiste en evaluar estos resultados con la expresión  $t > t_{\alpha}$ .

- Servicio TwitterCalls:  $t = 4,98$  ;  $t_{\alpha=0.025} = 2,7765$ , entonces  $t > t_{\alpha}$
- Servicio TwitterSMS:  $t = 4,11$  ;  $t_{\alpha=0.025} = 2,7765$ , entonces  $t > t_{\alpha}$
- Servicio CallMessage:  $t = 5,98$  ;  $t_{\alpha=0.025} = 2,7765$ , entonces  $t > t_{\alpha}$

Teniendo en cuenta los resultados obtenidos, se puede concluir: La Hipótesis nula  $H_0$  se rechaza, y se acepta la Hipótesis alternativa  $H_a$  con un porcentaje de 97,5% de probabilidad, de que si el experimento se vuelve a realizar bajo las mismas condiciones planteadas, la composición de los servicios convergentes CallMessage, TwitterSMS y CallMessage mediante patrones de flujo de ejecución reduce el tiempo de composición en comparación al tiempo empleado cuando no se utilizan patrones.

Finalmente, con los resultados de las pruebas **T1** y **T2**, se puede afirmar: la composición de servicios convergentes mediante el uso de patrones de flujo de ejecución reduce el tiempo de composición en comparación al tiempo empleado para componer servicios a nivel de código.

## 5.3 Resumen

Este Capítulo presentó el prototipo funcional del presente trabajo de grado, el cual se basa en la implementación de una herramienta gráfica de composición de servicios Telco 2.0 basada en patrones de flujo de ejecución.

Al principio de este Capítulo se describieron los componentes de la herramienta, los diagramas de despliegue, orquestación, paquetes, subsistemas y capas del sistema; los cuales son necesarios para dar soporte a la composición de servicios Telco 2.0 mediante patrones de flujo de ejecución.

Finalmente, se mostraron los resultados de las pruebas realizadas con 15 sujetos de prueba, los cuales fueron analizados mediante estadística descriptiva e inferencial, lo cual permitió demostrar que el proceso de composición de servicios convergentes mediante el uso de patrones de flujo de ejecución en un entorno Telco 2.0, reduce considerablemente el tiempo de dicho proceso, en comparación cuando se realiza sin patrones.

## Capítulo 6

### 6 Conclusiones, resultados y trabajo futuro

En este Capítulo son presentadas las conclusiones a las cuales se llegó con el desarrollo de este trabajo de grado, los resultados obtenidos y finalmente se proponen algunos trabajos futuros que se podrían realizar teniendo como base el presente trabajo.

#### 6.1 Conclusiones

A continuación se exponen las principales conclusiones del desarrollo del presente trabajo de grado.

- Se realizó un análisis de las investigaciones y trabajos relacionados con la composición de servicios convergentes, algunos de estos con el soporte de patrones de flujo de ejecución, en los cuales se encontró una carencia de modelos que implementen y ejecuten directamente dichos patrones en entornos convergentes como JAIN SLEE.
- Actualmente existe una gran variedad de servicios en entornos convergentes, debido a esto se propuso un catálogo de servicios Telco, Web y convergentes, el cual permitió realizar una clasificación de las tendencias y servicios utilizados por organizaciones de estandarización (3GPP [47], ITU [48], ETSI-TISPAN [50], OneAPI [51]), plataformas para despliegue de servicios (Mobicents [32], Rhino [31]) y algunos operadores de telecomunicaciones (Vodafone, Tigo, Claro).
- El presente trabajo de grado presentó el estudio de los patrones de flujo de ejecución soportados dentro del proceso de composición de servicios convergentes. Este estudio se realizó mediante la detección de los patrones de flujo de control más recurrentes en treinta servicios (Web, Telco, convergentes) y de los patrones de datos soportados por la especificación JAIN SLEE. De esta forma se obtuvo como resultado un total de 13 CFP y 16 DP que permiten soportar la creación y composición de servicios convergentes mediante el uso de patrones de flujo de ejecución en un entorno Telco 2.0.
- La detección de patrones de flujo de control se basa en la utilización de dos representaciones formales: redes de Petri y grafos. La primera describe el flujo de

control a través de nodos dirigidos y compuertas lógicas; y la segunda, brinda una descripción más específica de los servicios y permite realizar una asociación entre los servicios presentes en el catálogo y los patrones de flujo de control y datos propuestos por Vander Aalst *et al.*[7].

- El método propuesto por el proyecto de investigación, dio como resultado inicial un conjunto de 21 patrones de flujo de control, a los cuales se les realizó un análisis de recurrencia en servicios Telco 2.0, obteniendo como resultado final 13 patrones comunes presentes de dichos servicios, los cuales se ubican en las categorías de: patrones básicos, basados en estados, de cancelación y patrones estructurales [7]. Por otra parte, el análisis realizado a los patrones de datos dio como resultado un conjunto de 16 patrones de datos, relacionados con las capacidades del entorno de ejecución de servicios JAIN SLEE.
- Se utilizó la tecnología JAIN SLEE, la cual se basa en una especificación abierta y estándar para un servidor de aplicación de telecomunicaciones diseñado para altos niveles de desempeño en aplicaciones orientadas a eventos, y que permite el reúso de componentes. Adicionalmente, permite a través de su arquitectura de adaptadores de recursos, realizar una abstracción de nivel de control de un entorno Telco 2.0, logrando un rápido desarrollo y despliegue de nuevas aplicaciones y servicios.
- Las pruebas realizadas a 15 estudiantes de la Universidad del Cauca con experiencia en programación orientada a objetos y JAIN SLEE, los cuales crearon 3 servicios Telco 2.0, algunos utilizando la herramienta Gt-4SC y otros utilizando sus conocimientos de programación; deja un banco de pruebas con 60 tiempos de composición de servicios. Dichos tiempos fueron analizados mediante estadística descriptiva e inferencial, lo cual permite concluir que la composición de servicios convergentes mediante el uso de patrones de flujo de ejecución, es más eficiente que realizar el mismo proceso sin patrones.
- Los resultados obtenidos en las pruebas realizadas, permiten concluir que la implementación de patrones de flujo de ejecución, directamente en JAIN SLEE y la reutilización de componentes con soporte gráfico, facilitan la composición de servicios convergentes en un entorno Telco 2.0 y reducen el tiempo de este proceso; facilitando así el trabajo de personas expertas relacionadas en este campo.

## 6.2 Resultados

A continuación se presentan los resultados que dejó el trabajo de grado después de su finalización.

- Un catálogo de servicios presentes en un entorno Telco 2.0, el cual comprende un total de 73 servicios (23 Telco, 38 Web, 12 convergentes y servicios de seguridad y acceso a datos) obtenidos mediante el estudio de organizaciones de estandarización (3GPP [47], ITU [48], ETSI-TISPAN [50], OneApi [51]), plataformas para despliegue de servicios (Mobicents [32], Rhino [31]) y algunos operadores de telecomunicaciones (Vodafone, Tigo, Claro).

- Herramienta grafica nombrada: **GT-4SC** (Graphical Tool-patterns-based for Service Creation), la cual es una extensión para Eclipse, desarrollado e implementado en este trabajo de investigación. GT-4SC permite la creación de nuevos servicios convergentes por medio de módulos reutilizables. Cada módulo tiene un código fuente asociado, que se genera automáticamente al agregar uno de estos al área de trabajo y permite al desarrollador, componer nuevos servicios convergentes evitando el contacto con el código fuente y la configuración de archivos complementarios de funcionamiento. GT-4SC es una herramienta para la composición de servicios convergentes basada en patrones de flujo de ejecución, lo cual permite al usuario, implementar nuevos servicios teniendo en cuentas las perspectivas de flujo de control y datos.
- Un banco de pruebas el cual se desarrolló utilizando 15 estudiantes de Ingeniería Electrónica y Telecomunicaciones de último semestre, próximos a obtener su título y estudiantes de Maestría en Ingeniería Telemática, con conocimientos en programación y JAIN SLEE.
- Aporte al trabajo de maestría en Ingeniería Telemática del ingeniero Jesús David Ramírez titulado: Orquestación Automática de Servicios Convergentes.
- **Publicación:** “Control-Flow Patterns in Converged Services” presentado en la conferencia: “Service Computation 2012: The Fourth International Conferences on Advanced Service Computing” realizada en la ciudad de Niza, Francia. Artículo publicado en Agosto de 2012 con Copyright (c) IARIA, 2012. ISBN: 978-1-61208-215-8.

## 6.3 Trabajo Futuro

- La iniciativa de Workflow Patterns define cuatro perspectivas: datos, control, recursos y manejo de excepciones [61]. Para el presente trabajo de grado, se realizó un estudio de las dos primeras. Seria de especial interés incluir todas las perspectivas para la composición de servicios convergentes.
- Un entorno Telco 2.0 está compuesto por dos capas principales: la capa de aplicación y control. Para la implementación de estas capas se utilizó la especificación JAIN SLEE, sin embargo, se propone como trabajo futuro incluir la arquitectura IMS en la capa de control del prototipo funcional, la cual sería la encargada de recibir toda la señalización proveniente de otras redes.
- Para el presente trabajo de grado, se tuvo en cuenta las evaluaciones realizadas por Vander Aalst *et al.* para los lenguajes jBPM y BPEL, los cuales son utilizados en los trabajos [4] [5] [6] para la composición de servicios convergentes. Se plantea como trabajo futuro, el análisis de otras evaluaciones para otros lenguajes realizadas por Vander Aalst en [7].
- El estudio realizado de los patrones de flujo de ejecución para el presente trabajo de grado, permite componer servicios de forma estática, sin embargo, seria de especial

interés estudiar estos patrones para composición de servicios de manera automática o dinámica.

- La herramienta GT-4SC, permite agregar o quitar componentes en un espacio de trabajo de forma manual, esta capacidad permite crear servicios convergentes de forma sencilla e intuitiva; por lo anterior, se plantea como trabajo futuro, incluir capacidades a la herramienta que permita reconfigurar los servicios de forma automática, en caso de que alguno de los componentes falle, sin necesidad de que el usuario tenga que retirarlo y agregarlo de forma manual al área de trabajo para que el servicio vuelva a funcionar.
- Para el estudio de los patrones presentes en servicios de un entorno Telco 2.0, se realizó el modelado de 30 servicios, utilizando las representaciones formales de redes de Petri y Grafos, sin embargo, estudios posteriores pueden aumentar el número de servicios modelados, con el fin de determinar si se aumenta la cantidad de patrones detectados.
- Para el prototipo funcional, la herramienta GT-4SC desarrollada en el marco del presente trabajo de grado, incluye un conjunto de seis servicios básicos (Web y Telco). Como trabajo futuro se propone ampliar el número de servicios básicos implementados, lo cual permitiría el acceso a funcionalidades adicionales ofrecidas por servicios Web como Google, Facebook, etc. y servicios de telecomunicaciones como Text to Speech, buzón de mensajes; entre muchos otros.

## 7 Referencias

- [1] P. Falcarin and C. Venezia, "Communication Web Services and JAIN-SLEE Integration Challenges," *International Journal of Web Services Research*, vol. 5, no. 4, pp. 59–78, 2008.
- [2] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," in *International Workshop on Semantic Web Services and Web Process Composition, SWSWPC*, 2004, pp. 43 – 54.
- [3] A. Mendonça, A. Bhayan, B. Baranowski, D. Silas, E. Martins, I. Ivanov, J. Morgan, and C.-M. Bjarkell, "JAIN SLEE User Guide - Mobicents," 2012. [Online]. Available: [http://docs.jboss.org/mobicents/jain-slee/2.7.0.FINAL/container/user-guide/en-US/pdf/Mobicents\\_SLEE\\_Container\\_User\\_Guide.pdf](http://docs.jboss.org/mobicents/jain-slee/2.7.0.FINAL/container/user-guide/en-US/pdf/Mobicents_SLEE_Container_User_Guide.pdf).
- [4] T. Eichelmann, W. Fuhrmann, U. Trick, and B. Ghita, "Enhanced concept of the TeamCom SCE for automated generated services based on JSLEE," in *Eighth International Network Conference (INC )*, 2010.
- [5] M. Femminella, E. Maccherani, and G. Reali, "A software architecture for simplifying the JSLEE service design and creation," in *Software, Telecommunications and Computer Networks (SoftCOM)*, 2010, vol. 22.
- [6] D. Zhu, Y. Zhang, B. Cheng, B. Wu, and J. Chen, "HSCEE: A Highly Flexible Environment for Hybrid Service Creation and Execution in Converged Networks," *Journal of Convergence Information Technology*, vol. 6, no. 3, pp. 264–276, 2011.
- [7] W. M. P. van der Aalst, A. Hofstede, N. Russell, and N. Mulyar, "Workflow Control-Flow Patterns, A Revised View," in *BPM Center Report BPM-06-22*, 2006.
- [8] W. M. P. van der Aalst, N. Russell, A. Hofstede, and D. Edmond, "Workflow Data Patterns," *QUT Technical report, FIT-TR-2004-01*, 2004.
- [9] J.-L. Yoon, "Telco 2.0: a new role and business model," *Communications Magazine, IEEE*, vol. 45, no. 1, pp. 10–12, 2007.
- [10] STL Partners, "Telco 2.0 - Business Model Innovation for the Digital Economy," 2012. [Online]. Available: [http://www.stlpartners.com/telco2\\_index.php](http://www.stlpartners.com/telco2_index.php).

- [11] J. Ortiz, "Pymes y telecomunicaciones, dupla que deriva en competitividad," 2011. [Online]. Available: <http://www.manejodecrisis.com.mx/index.php/acervo/colaboraciones-en-medios/762-pymes-y-telecomunicaciones-dupla-que-deriva-en-competitividad>.
- [12] J. González, C. A. Iglesias, and F. Echanique, "La Plataforma Telcoblocks de Despliegue y Desarrollo de Servicios VoIP," 2009. [Online]. Available: <http://www.gsi.dit.upm.es/index.php/es/component/jresearch/?task=show&view=publication&id=204>.
- [13] J. Niemöller, E. Freiter, K. Vandikas, R. Quinet, R. Levenshteyn, and I. Fikouras, "Composition in Converged Service Networks: Requirements and Solutions Background and State of the Art," in *Workshop on Business System Management and Engineering (BSME)*, 2010.
- [14] W. M. P. van der Aalst, "Don't go with the flow: Web services composition standards exposed," *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 72–76, 2003.
- [15] W. M. P. van der Aalst, N. Trcka, and N. Sidorova, "Analyzing Control-Flow and Data-Flow in Workflow Processes in a Unified Way," *Computer Science, Technische Universiteit Eindhoven, The Netherlands*, vol. 8, no. 31, pp. 2–7, 2008.
- [16] L. H. Thom, C. Iochpe, and M. Reichert, "Workflow Patterns for Business Process Modeling," *BPMDS*, vol. 4495, pp. 349–357, 2007.
- [17] Oracle- Community Development of java Technology Specifications, "JSR 240: Jain Slee v 1.1," 2012. [Online]. Available: <http://www.jcp.org/en/jsr/detail?id=240>.
- [18] Y. J. Byun, B. A. Sanders, and C. S. Keum, "Design patterns of communicating extended finite state machines in sdl," in *8th Conference on Pattern Languages of Programs (PLoP'01)*, 2001.
- [19] J. Yu, Q. Z. Sheng, and P. Falcarin, "A Visual Semantic Service Browser Supporting User-Centric Service Composition," in *24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 244–251.
- [20] O. Droegehorn, E. Kovacs, J. Cupillard, G. LeJeune, M. Belaunde, and I. Koenig, "Professional and End-User driven Service Creation in the SPICE platform," in *IEEE\_WOWMOM*, 2008.
- [21] Y. J. Sun, J. W. Cui, and Y. J. Gu, "Implementing advanced workflow patterns in WebSphere Integration Developer and WebSphere Process Server , Part 1: Basic control flow patterns and cancellation and force completion patterns." pp. 1–31, 2010.
- [22] M. E. Mejía and P. Álvarez, "El ciclo de vida de un servicio Web compuesto: virtudes y carencias de las soluciones actuales," 2005. [Online]. Available: <http://webdiis.unizar.es/~mespinoz/ReporteComposición.pdf>.



- [23] K. Al-Begain, C. Balakrishna, L. A. Galindo, and D. Moro, *IMS: A Development and Deployment Perspective*. Chichester, West Sussex, U.K.: , 2009, pp. 220–270.
- [24] E.-P. Lim, C.-H. Lee, and C.-H. Chen, “Dynamic Web Service Selection for Reliable Web Service Composition,” *Service Computing, IEEE Transactions*, vol. 1, no. 2, pp. 104–116, 2008.
- [25] G. Bond, E. Cheung, R. Levenshteyn, and Loloannis Fikouras, “Unified Telecom and Web Services Composition: Problem Definition and Future Directions,” in *Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications*, 2009.
- [26] 3GPP, “Parlay X Web Services; Part: 1 Common,” 2009.
- [27] Open Cloud, “A Slee for all Seasons,” 2006. [Online]. Available: <http://www.opencloud.com/documents/Whitepaper A SLEE for all Seasons.pdf>.
- [28] F. Telbisz and B. Gódor, “Service Oriented Architectures for convergent Service Delivery Platforms,” 2006. [Online]. Available: <http://archive.eurescom.eu/~pub/deliverables/documents/P1600-series/P1652/D1/P1652-D1.pdf>.
- [29] Ericsson, “Ericsson Service Delivery Platform,” 2005. [Online]. Available: <http://archive.ericsson.net/service/internet/picov/get?DocNo=28701-FGB101265>.
- [30] W. M. P. van der Aalst, A. Hofstede, B. Kiepuszewski, and A. Barros, “WorkFlow Patterns,” *Distributed and Parallel Databases*, vol. 14, pp. 5–51, 2003.
- [31] Open Cloud, “Rhino’s SIP resource adaptor (RA), services and components.” [Online]. Available: <https://developer.opencloud.com/devportal/display/RD/1+About+the+Sample+Applications>. [Accessed: 08-Mar-2012].
- [32] Mobicents, “Mobicents Examples.” [Online]. Available: <http://www.mobicents.org/examples.html>.
- [33] ITU, “NGN service requirements and capabilities for network aspects of applications and services using tag-based identification,” *Recomendación ITU-T Y.2213*, 2008. [Online]. Available: <http://www.itu.int/rec/T-REC-Y.2213-200809-I>.
- [34] I. G. Ben Yahia, E. Bertin, J. P. Deschrevel, and N. Crespi, “Service Definition for Next Generation Networks,” in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL)*, 2006, pp. 22–29.
- [35] Ministerio de las Tecnologías de la Información y Comunicaciones - Decreto 2870 - 2007, “Servicios de valor agregado.” [Online]. Available: <http://www.mintic.gov.co>.

- [36] Comisión nacional de comunicaciones de Argentina, “Resolución CNT 1083/95.” [Online]. Available: <http://www.cnc.gov.ar/infotecnica/sva/index.asp>.
- [37] ITU – T, “NGN FG Proceedings - Part II,” 2005. [Online]. Available: <http://www.itu.int/en/ITU-T/gsi/ngn/Pages/release1.aspx>.
- [38] ITU-T, “Vocabulario de términos relativos a la conmutación y la señalización,” 1988. [Online]. Available: <http://www.itu.int/rec/T-REC-Q.9-198811-I>.
- [39] Ministerio de las tecnologías de la información y Comunicaciones, “Glosario de términos y definiciones del Sector de Telecomunicaciones.” [Online]. Available: <http://www.mintic.gov.co/index.php/glosario>.
- [40] 3GPP, “3GPP TS 27.173 - Multimedia Telephony Service and supplementary services,” 2010.
- [41] W. Reichl and E. O. Ruhle, “Competition and Services in Next Generation Networks,” in *17th Biennial International Telecommunications Society (ITS)*, 2008.
- [42] R. Taylor, B. Zhang, and S. Chen, “Value Added Services Policy Reform in China: Lessons for and from the US in Managing an Evolving Market,” in *Telecommunications Policy Research Conference*, 2007.
- [43] ITU-T, “Definition of Teleservices , ITU T Recommendation I.240.” 1989.
- [44] ITU-T, “Definition of Supplementary Services, ITU T Recommendation I.250.” 1988.
- [45] S. Shang, E. Li, Y. Wu, and O. Hou, “Understanding Web 2.0 service models: A knowledge-creating perspective,” *Information & Management*, vol. 48, no. 4–5, pp. 178–184, May 2011.
- [46] Ajaxprojects.com, “Web 2.0 - Ajax Projects.” [Online]. Available: <http://web2.ajaxprojects.com/web2/>.
- [47] 3GPP, “3GPP Global initiative – about 3GPP.” [Online]. Available: <http://www.3gpp.org/About-3GPP>.
- [48] ITU, “ITU Web Site.” [Online]. Available: <http://www.itu.int/en/ITU-T/publications/Pages/recs.aspx>.
- [49] ITU – T, “Telecommunication Standardization Sector.” [Online]. Available: <http://www.itu.int/en/publications/ITU-T/Pages/default.aspx>.
- [50] ETSI-TISPAN Wen site, “ETSI-TISPAN.” [Online]. Available: <http://www.etsi.org/WebSite/Standards/ETSIDeliverables.aspx>.
- [51] One Api, “OneApi Rerefence Implementatio.” [Online]. Available: <http://oneapi.aepona.com/>.

- [52] I. Badr and M. Belaunde, "TelcoML - UML Profile for Integrated Telecom," 2012. [Online]. Available: <http://www.tmforum.org/Community/groups/liaison/downloads.aspx>.
- [53] Rhino Developers, "Rhino Slee Compendium." [Online]. Available: <https://developer.opencloud.com/devportal/display/OCDEV/JSLEE+Application+Compendium>.
- [54] Tigo Colombia, "Servicios prestados por tigo colombia." [Online]. Available: <http://www.tigo.com.co>.
- [55] E. Martin - Jboss Comunity, "Mobicents JAIN SLEE JDBC Resource Adaptor User Guide." [Online]. Available: [http://docs.jboss.org/mobicents/jain-slee/2.5.0.FINAL/resources/jdbc/user-guide/en-US/html\\_single/](http://docs.jboss.org/mobicents/jain-slee/2.5.0.FINAL/resources/jdbc/user-guide/en-US/html_single/).
- [56] CSIC, "Departamento de tratamiento de la informacion y codificación." [Online]. Available: <http://www.iec.csic.es/>.
- [57] P. Asadoorian, "An Introduction to IPsec." [Online]. Available: <http://pauldotcom.com/IPSEC>. [Accessed: 25-Feb-2012].
- [58] gnu.org, "The GNU Transport Layer Security Library." [Online]. Available: <http://www.gnu.org/software/gnutls/documentation.html>. [Accessed: 22-Feb-2012].
- [59] Network Working Group, "HTTP Over TLS - RFC 2818." [Online]. Available: <http://tools.ietf.org/html/rfc2818>. [Accessed: 29-Feb-2012].
- [60] P. Wohed, B. Andersson, A. H. Hofstede, N. Russell, and W. M. P. van der Aalst, "Patterns-based Evaluation of Open Source BPM Systems: The Cases of jBPM , OpenWFE , and Enhydra Shark," *Information and Software Technology*, vol. 51, no. 8, pp. 1187– 1216, 2009.
- [61] "Workflow Patterns Initiative." [Online]. Available: <http://www.workflowpatterns.com/>. [Accessed: 03-Mar-2012].
- [62] J. L. Peterson, *Petri net theory and the modeling of systems*. Prentice Hall, 1981.
- [63] J. Paananen, "Introduction to and comparison of formalisms." [Online]. Available: <http://www.tml.tkk.fi/Opinnot/Tik-110.501/1995/intfo.html>.
- [64] B. Hendrickson and R. Leland, "A multilevel algorithm for partitioning graphs," in *ACM/IEEE conference on Supercomputing*,, 1995.
- [65] C. Figueroa, "Descubrimiento automático de procesos de negocio basado en semántica de comportamiento," Tesis de Maestria - Universidad del Cauca, 2011.
- [66] A. Spiteri, "Rewriting Petri Nets as Directed Graphs," *International Journal of Computers*, vol. 5, no. 2, 2011.

- [67] C. Corrales, J. E. Daza, and J. C. Corrales, *Comparacion Estructural y Linguistica de Procesos de Negocio Semanticos*. Editorial - Universidad del Cauca, 2012.
- [68] P. Foggia, C. Sansone, and M. Vento, "A Performance Comparison of Five Algorithms for Graph Isomorphism," in *3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, 2001, pp. 188–199.
- [69] L. P. Cordella, C. Sansone, and M. Vento, "Graph Isomorphism Algorithm for Matching Large Graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1367 – 1372, 2004.
- [70] A. Ferro, R. Giugno, M. Mongioví, A. Pulvirenti, D. Skripin, and D. Shasha, "GraphBlast: multi-feature graphs database searching," in *NETTAB 2007 Workshop*, 2007.
- [71] L. P. Cordella, P. Foggia, C. Sansone, and D. Skripin, "Netmatch is a cytoscape plugin to query networks for patterns," *Bioinformatics*, vol. 23, no. 7, pp. 910 – 912, 2007.
- [72] D. Rivas, D. Corchuelo, C. Figueroa, J. Corrales, and R. Giugno, "Business Process Model Retrieval based on Graph Indexing Method," *Lecture Notes in Business Information Processing*, vol. 66 Part 3, pp. 238–250, 2011.
- [73] WorkFlow patterns, "Workflow data patterns - Evaluation," 2006. [Online]. Available: <http://www.workflowpatterns.com/evaluations/>.
- [74] Jboss Community, "Mobicents JAIN SLEE Datasources." [Online]. Available: <https://community.jboss.org/wiki/MobicentsJAINSLLEEDatasources>. [Accessed: 14-Oct-2012].
- [75] Eclipse ORG, "Eclipse Downloads," 2012. [Online]. Available: <http://www.eclipse.org/downloads/>.
- [76] Twitter for Developers, "OAuth for Twitter," 2012. [Online]. Available: <https://dev.twitter.com/docs/auth>.
- [77] OAuth.Net Framework, "OAuth," 2012. [Online]. Available: <http://oauth.net/>.
- [78] OAuth Working Group, "The OAuth 2.0 Authorization Framework - draft-ietf-oauth-v2-31," 2012.
- [79] Twitter4J Support team, "Twitter for java Twitter4j," 2011. [Online]. Available: <http://twitter4j.org/en/index.jsp>.
- [80] Mobicents, "Mobicents JAINSLLEE Downloads," 2012. [Online]. Available: <http://www.mobicents.org/slee/downloads.html>.
- [81] Mobicents, "Mobicents EclipSlee," 2012. [Online]. Available: <http://sourceforge.net/projects/mobicents/files/>.

- [82] RedHat - Jboss Enterprise, "JBoss AS - Installation and getting Started." [Online]. Available: [http://docs.jboss.org/jbossremoting/2.5.3.SP1/html\\_single/](http://docs.jboss.org/jbossremoting/2.5.3.SP1/html_single/).
- [83] D. M. López, B. Blobel, and C. González, "Quality Evaluation of Health Information System's Architectures Developed Using the HIS-DF Methodology," in *Medical and Care Compunetics* 6, 2010, pp. 21 –31.
- [84] H. Sierra, *Fundamentos de estadística y análisis de datos en SPSS - Grupo de Investigación en Genética Humana*. Editorial - Universidad del Cauca, 2011.
- [85] W. Mendenhall, R. Beaver, and B. Beaver, "Introducción a la Probabilidad Y Estadística - Inferencias a Partir de pequeñas muestras." Cenage Learning, pp. 328 – 400, 2008.
- [86] N. Russell, A. H. ter Hofstede, and W. M. P. van der Aalst, "newYAWL: Specifying a Workflow Reference Language using Coloured Petri Nets," in *Eighth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools - CPN'0*, 2007.