

**RECUPERACIÓN AUTOMÁTICA DE SERVICIOS EN ENTORNOS  
CONVERGENTES A PARTIR DE SOLICITUDES REALIZADAS EN LENGUAJE  
NATURAL**



Universidad  
del Cauca

**EDGAR CAMILO PEDRAZA ALARCÓN  
JULIÁN ANDRÉS ZÚÑIGA GALLEGO**

Monografía presentada para optar al título de Ingeniero en  
Electrónica y Telecomunicaciones

Director: PhD. Ing. Juan Carlos Corrales Muñoz

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE TELEMÁTICA  
POPAYÁN  
2012**

# TABLA DE CONTENIDO

Pag.

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.1. CONTEXTO DE INVESTIGACIÓN.....</b>	<b>1</b>
<b>1.2. MOTIVACIÓN.....</b>	<b>2</b>
<b>1.3. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>3</b>
<b>1.4. OBJETIVOS DE INVESTIGACIÓN.....</b>	<b>4</b>
1.4.1. Objetivo general.....	4
1.4.2. Objetivos específicos .....	4
1.4.3. Hipótesis de Investigación .....	5
<b>1.5. ENFOQUE .....</b>	<b>5</b>
<b>1.6. MODELO CONCEPTUAL .....</b>	<b>7</b>
<b>1.7. ALCANCE .....</b>	<b>7</b>
<b>1.8. CONTRIBUCIONES.....</b>	<b>8</b>
<b>1.9. CONTENIDO DE LA MONOGRAFÍA .....</b>	<b>10</b>
<b>2. ESTADO DEL ARTE .....</b>	<b>11</b>
<b>2.1. CONCEPTOS GENERALES .....</b>	<b>11</b>
2.1.1. Entorno Convergente.....	11
2.1.2. Recuperación de Servicios en la Web .....	11
2.1.3. Recuperación de Servicios de Telecomunicaciones .....	12
2.1.4. Representación de Conocimiento.....	12
2.1.5. Procesamiento de Lenguaje Natural .....	14
<b>2.2. TRABAJOS RELACIONADOS.....</b>	<b>16</b>
2.2.1. Recuperación de Servicios .....	17
2.2.2. Procesamiento de Lenguaje Natural .....	19
2.2.3. Brechas Existentes.....	20
<b>2.3. RESUMEN .....</b>	<b>22</b>
<b>3. ANÁLISIS DE LENGUAJE NATURAL.....</b>	<b>23</b>
<b>3.1. PROCESAMIENTO DE LENGUAJE NATURAL.....</b>	<b>23</b>
3.1.1. Modelo Conceptual .....	24
3.1.2. Niveles de análisis lingüístico.....	25
<b>3.2. PLATAFORMAS PARA EL PROCESAMIENTO DE LENGUAJE NATURAL .....</b>	<b>31</b>
3.2.1. Apache UIMA ( <i>Unstructured Information Management Applications</i> ).....	31
3.2.2. OpenNLP.....	32
3.2.3. GATE ( <i>General Architecture for Text Engineering</i> ).....	32
3.2.4. NLTK (Natural Language Toolkit) .....	33
3.2.5. Discusión. ....	33
<b>3.3. EJEMPLO.....</b>	<b>34</b>

3.4.	RESUMEN .....	36
4.	RECUPERACIÓN DE SERVICIOS ORIENTADO AL USUARIO FINAL .....	37
4.1.	SOLUCIÓN A LIMITANTES DE VOCABULARIO.....	38
4.1.1.	Extracción semántica del vocabulario del usuario .....	38
4.1.2.	Enriquecimiento semántico al vocabulario del usuario.....	39
4.2.	RECUPERACIÓN DE SERVICIOS A PARTIR DE LA SOLICITUD DEL USUARIO.....	43
4.2.1.	Sistema de Recuperación de Servicios .....	43
4.3.	RECUPERACIÓN DE SERVICIOS CON BASE EN PROPIEDADES NO FUNCIONALES .....	50
4.3.1.	Propiedades No Funcionales (PNF).....	50
4.3.2.	Selección de parámetros No Funcionales de los servicios. ....	51
4.3.3.	Ranking de Servicios con PNF.....	51
4.4.	OBTENCIÓN FINAL DEL RANKING DE SERVICIOS .....	54
4.4.1.	Factor de Afinación. ....	54
4.4.2.	Flujo de control genérico. ....	54
4.4.3.	Recomendación del Flujo de Control genérico. ....	56
4.5.	EJEMPLO.....	58
4.6.	RESUMEN .....	60
5.	PROTOTIPO Y EVALUACIÓN.....	61
5.1.	DESCRIPCIÓN DEL PROTOTIPO.....	61
5.1.1.	Modelo de Casos de Uso del Sistema .....	61
5.1.2.	Descripción de la Arquitectura de Referencia.....	63
5.2.	PRUEBAS DEL PROTOTIPO .....	68
5.2.1.	Módulo de Procesamiento de Lenguaje Natural .....	69
5.2.2.	Módulo de Recuperación de Servicios.....	70
5.2.3.	Plan de Pruebas y Resultados Obtenidos .....	74
5.3.	RESUMEN .....	88
6.	CONCLUSIONES Y TRABAJO FUTURO.....	90
6.1.	CONTRIBUCIONES.....	90
6.2.	CONCLUSIONES.....	91
6.3.	TRABAJOS FUTUROS .....	93

## INDICE DE FIGURAS

	Pág.
<i>Figura 1. Aplicación creada usando Yahoo! Pipes.....</i>	<i>3</i>
<i>Figura 2. Enfoque de Investigación del Trabajo de Grado.....</i>	<i>5</i>
<i>Figura 3. Modelo Conceptual Básico.....</i>	<i>7</i>
<i>Figura 4. Modelo de una Solicitud en Lenguaje Natural.....</i>	<i>24</i>
<i>Figura 5. Modelo Conceptual.....</i>	<i>24</i>
<i>Figura 6. Flujo conceptual del proceso de recuperación de servicios.....</i>	<i>37</i>
<i>Figura 7. Sistema General de Recuperación de Servicios.....</i>	<i>44</i>
<i>Figura 8. Modelo de la descripción de un Servicio.....</i>	<i>46</i>
<i>Figura 9. Similitud de Vectores.....</i>	<i>57</i>
<i>Figura 10. Flujo final de servicios.....</i>	<i>59</i>
<i>Figura 11. Diagrama de Casos de Uso del Sistema.....</i>	<i>63</i>
<i>Figura 12. Arquitectura de Referencia del Sistema.....</i>	<i>65</i>
<i>Figura 13. Gráfica de Rendimiento - Recuperación de términos relevantes de la solicitud en lenguaje natural.....</i>	<i>76</i>
<i>Figura 14. Acuerdo de clasificación de términos.....</i>	<i>77</i>
<i>Figura 15. Acuerdo de estructura de solicitud formal.....</i>	<i>77</i>
<i>Figura 16. Validación de modulo.....</i>	<i>78</i>
<i>Figura 17. Medida de Desempeño: Precisión y Recall para la tarea de Desambiguación.....</i>	<i>78</i>
<i>Figura 18. Medida de Desempeño: Precisión y Recall para la tarea de Reconocimiento de Nombres de Entidad.....</i>	<i>79</i>
<i>Figura 19. Medida de Desempeño: Precision y Recall para la tarea de Etiquetado de Palabras.....</i>	<i>80</i>
<i>Figura 20. Gráfica de rendimiento del sistema en la recuperación de servicios respecto al número de términos recuperados de la solicitud en lenguaje natural.....</i>	<i>81</i>
<i>Figura 21. Gráfica de rendimiento del sistema en la recuperación de servicios respecto al número de servicios disponibles.....</i>	<i>82</i>
<i>Figura 22. Medidas de Desempeño: Precision para sistemas de recuperación de servicios.....</i>	<i>83</i>
<i>Figura 23. Medidas de Desempeño: Recall para sistemas de recuperación de servicios.....</i>	<i>83</i>
<i>Figura 24. Medidas de Desempeño: Overall para sistemas de recuperación de servicios.....</i>	<i>84</i>
<i>Figura 25. Gráfica de Precision vs. Recall para la recuperación de servicios.....</i>	<i>84</i>
<i>Figura 26. Medidas de Desempeño: Ganancia Acumulada Normalizada Media (ANCG) para sistemas de recuperación de servicios.....</i>	<i>85</i>
<i>Figura 27. Medidas de Desempeño: Ganancia Acumulada Descontada Normalizada Media (ANDCG) para sistemas de recuperación de servicios.....</i>	<i>85</i>
<i>Figura 28. Medidas de Desempeño: Generalización de Precisión Fija Promedio (GenAveP') para sistemas de recuperación de servicios.....</i>	<i>86</i>
<i>Figura 29. Medidas de Desempeño: Precision para el sistema de refinamiento de ranking.....</i>	<i>87</i>
<i>Figura 30. Medidas de Desempeño: Recall para el sistema de refinamiento de ranking.....</i>	<i>87</i>
<i>Figura 31. Medidas de Desempeño: Overall para el sistema de refinamiento de ranking.....</i>	<i>88</i>
<i>Figura 32. Medidas de Desempeño: Ganancia Acumulada Descontada Normalizada Media (ANDCG) para el sistema de refinamiento de ranking.....</i>	<i>88</i>

## INDICE DE TABLAS

	<b>Pág.</b>
<i>Tabla 1. Modelo Híbrido de Representación de Conocimiento, adaptado de [78].....</i>	<i>14</i>
<i>Tabla 2. Comparación de Propuestas de Trabajos Relacionados.....</i>	<i>21</i>
<i>Tabla 3. Comparación Herramientas para el procesamiento de Lenguaje Natural.....</i>	<i>34</i>
<i>Tabla 4. Comparación Herramientas para el procesamiento de Lenguaje Natural II .....</i>	<i>34</i>
<i>Tabla 5. Salida proceso de recuperación inicial de servicios. ....</i>	<i>58</i>
<i>Tabla 6. Salida proceso de recuperación de servicios con parámetros no funcionales.....</i>	<i>59</i>
<i>Tabla 7. Salida final proceso de recuperación de servicios.....</i>	<i>59</i>
<i>Tabla 8. Especificaciones del sistema. ....</i>	<i>66</i>
<i>Tabla 9. Plan de Pruebas.....</i>	<i>75</i>
<i>Tabla 10. Especificaciones Técnicas del Equipo empleado para las Pruebas del Prototipo.....</i>	<i>75</i>

# 1. INTRODUCCIÓN

## 1.1. CONTEXTO DE INVESTIGACIÓN

Con la digitalización de contenido, el cambio a redes basadas en IP, el acceso a ancho de banda de alta velocidad y la disponibilidad de comunicaciones multimedia, se ha logrado que arquitecturas de internet y telecomunicaciones evolucionen constantemente hacia una sola red, tanto en capas físicas y de transporte, como en la capa de servicios [1]. Esta evolución se ha evidenciado en la aparición de entornos convergentes, que permiten el acceso a aplicaciones basadas en la Web y a su vez, el consumo de servicios tradicionales de telecomunicaciones y nuevos de valor agregado [2]. Estos entornos prometen una “fácil” creación y despliegue de servicios, en contraste a lo que ofrecían las redes tradicionales, en las que el desarrollo de los servicios estaba limitado a la infraestructura de las tecnologías de red (conmutación de circuitos, por ejemplo: RTC- Red Telefónica Conmutada).

Actualmente, los entornos convergentes, pueden abarcar diferentes niveles de convergencia: red, servicios, industria, leyes, dispositivos y experiencia del usuario [2]. Siendo de especial interés para el desarrollo del presente proyecto, la convergencia de servicios y la experiencia del usuario, que buscan la creación de una interfaz única a los usuarios finales con tecnologías de telecomunicaciones, medios e informática, independiente del canal de comunicación. De esta forma, hasta el momento, se han adelantado esfuerzos para facilitar la recuperación de servicios, con el fin de crear servicios convergentes [3], basado en alguna descripción publicada de su funcionalidad y parámetros operacionales [5], ya que al ser una fase importante dentro del proceso de composición de nuevos servicios [4], da la capacidad a un sistema para encontrar y utilizar posteriormente un servicio.

En la recuperación de servicios es posible distinguir dos enfoques: la recuperación de servicios en la Web y en el dominio de las telecomunicaciones. La búsqueda de servicios Web puede basarse en mecanismos sintácticos o semánticos, donde los primeros utilizan la comparación de interfaces o de palabras claves [6], [7], lo que puede afectar el proceso con resultados de baja calidad, al no considerar las relaciones existentes entre las palabras de la solicitud [8]. Por su parte, los mecanismos semánticos permiten el establecimiento de relaciones entre conceptos que definen la funcionalidad de los servicios, obteniendo así una mayor precisión en la descripción de servicios y mejorando la calidad de los resultados, al recuperar servicios acordes a las necesidades del usuario [10], [11].

En cuanto a la recuperación de servicios en el dominio de las telecomunicaciones, con notables diferencias respecto a la búsqueda de servicios en la Web, es posible resaltar diferentes aproximaciones. Un claro ejemplo, son las redes de nueva generación, que ofrecen mecanismos de recuperación y composición de servicios, con el fin de proveer capacidades de red a terceras partes [1]. Otros enfoques basados en el protocolo SIP, permiten la identificación y recuperación de servicios a partir de una solicitud SIP sobre arquitecturas IMS [13], [14].

Aún con estos enfoques, la recuperación de servicios en entornos convergentes sigue siendo compleja; por ello, se consideran alternativas que ayuden a disminuir tal complejidad y hacer que el proceso de recuperación para el usuario sea más simple y eficiente.

Una de las alternativas utilizadas son las técnicas de Procesamiento de Lenguaje Natural (PLN), útiles para la obtención de información adicional a partir de solicitudes realizadas en lenguajes informales como el humano [15]. Estas técnicas han facilitado la comunicación hombre-máquina, ofreciendo interfaces y acceso a información cada vez más sofisticados; permitiendo a personas sin habilidades técnicas o de programación, especificar requerimientos de manera sencilla e interactuar de manera más fácil con

sistemas avanzados.

En general, los sistemas basados en técnicas de PLN, son capaces de procesar diferentes tipos de textos, articulados por seres humanos, para el entendimiento por parte de máquinas. Esto permite, por una parte, disminuir las brechas entre lenguajes informales (expresados como texto o voz) y lenguajes formales (estructurados p.ej. SPARQL [16]); y por otro lado, el acceso a información de manera más precisa y simple. Actualmente se desarrollan soluciones alrededor de sistemas de recuperación de información, máquinas de búsqueda y sistemas pregunta-respuesta, que utilizan diversos mecanismos y tareas de PLN.

Teniendo en cuenta los diferentes mecanismos disponibles para la ejecución del proceso de recuperación de servicios en entornos convergentes, el presente trabajo estudia las técnicas de recuperación utilizadas sobre servicios Web y otro tipo de recursos, que puedan ser adaptados al contexto de investigación planteado. Además, considera técnicas que permitan la disminución de la complejidad en el uso de sistemas de recuperación de servicios por parte de los usuarios finales, mejorando la experiencia de los mismos, tal como PLN.

## 1.2. MOTIVACIÓN

Una persona extranjera ha llegado a Colombia y planea comer los platos más deliciosos durante su estadía en el país, así que está buscando establecimientos de comidas y bebidas. Debido a su dificultad para hablar en español e interactuar con personas oriundas que puedan recomendarle lugares con platos exquisitos, ha traído consigo un Smartphone con la esperanza de crear una aplicación de manera simple que permita buscar información, la cual debe contener la palabra *comida*, tanto en el título como en la descripción del contenido. Además para decidir si es de su agrado o no, le gustaría ver imágenes de las comidas ofrecidas.

Formalmente, lo que el usuario desea hacer es buscar establecimientos de comidas y bebidas, y filtrar la información del contenido basado en el título y la descripción, los cuales deben contener la palabra *comida*. Adicionalmente, buscar dos imágenes de Flickr<sup>1</sup> que hayan sido etiquetadas con la palabra *comida*; para desarrollar la aplicación, el usuario usa herramientas basadas en interfaces orientadas a usuarios finales, como Yahoo! Pipes<sup>2</sup>. Para ello, como se muestra en la Figura 1, se necesitaron 7 componentes y configuraciones adicionales por parte del usuario, tales como la definición de la URL del sitio Web, gestionar las reglas de filtrado y definir el flujo de información entre componentes. Además, es indispensable que el usuario conozca la funcionalidad de los diferentes componentes; por ejemplo, el usuario debe conocer que Flickr es un sitio de gestión de imágenes. Es por esto que a pesar de ser una herramienta orientada al usuario final, debido a la complejidad y nivel de conocimiento requerido, esta aplicación es inapropiada para usuarios finales con poca experiencia sobre recursos como la Web.

---

<sup>1</sup> Disponible en <http://www.flickr.com/>

<sup>2</sup> Disponible en <http://pipes.yahoo.com/pipes/>

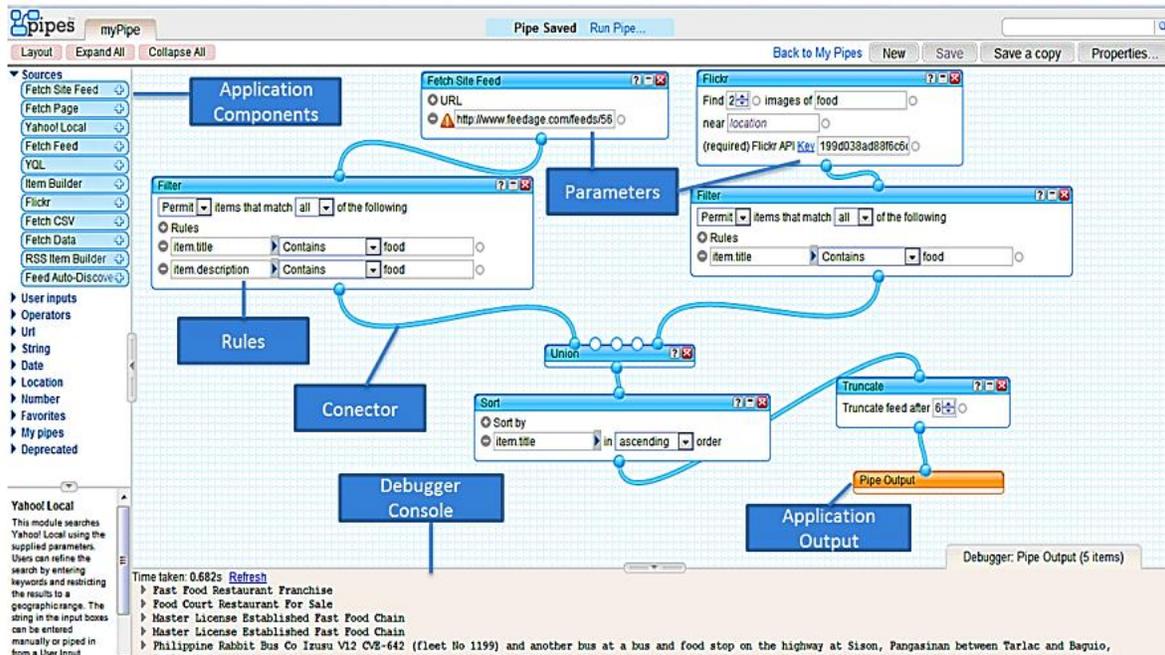


Figura 1. Aplicación creada usando Yahoo! Pipes

Con el incremento de repositorios de servicios Web online y desarrollo de servicios de telecomunicaciones, se requiere una alta precisión en la recuperación de servicios, pero también el desarrollo de sistemas intuitivos y fáciles de manejar por usuarios sin capacidades técnicas. Así, permitir que el usuario final realice una solicitud como: “*Quiero recibir imágenes de comida y lugares de venta de comida*”, simplifica notablemente las actividades de recuperación y organización de componentes, con el fin de realizar una funcionalidad semejante a la del sistema anterior. De esta forma, es posible lograr una reducción considerable de esfuerzo y recursos, mejorando indudablemente la interacción y la experiencia del usuario con el sistema. Es por esto que el objetivo principal del presente proyecto es brindar un sistema de recuperación de servicios con alta precisión y centrado en el usuario final.

### 1.3. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, el mundo de las TIC (Tecnologías de la Información y la Comunicación) ha cambiado notablemente debido a la Web 2.0, donde el usuario final, según su propia experiencia en Internet, no solo consume servicios o contenidos, sino que tiene la capacidad de generarlos, concepto denominado *Prosumer*. Este fenómeno ha constituido un cambio radical en la forma que las personas y las empresas perciben la información, modificando la manera en que éstas interactúan.

En consecuencia, la industria de las telecomunicaciones ha generado un nuevo modelo conocido como Telco 2.0 [17], el cual relaciona servicios y tecnologías de la Web 2.0 con los servicios de telecomunicaciones tradicionales. De esta manera, el paradigma Telco 2.0 abre la posibilidad de generar un novedoso portafolio de servicios, con el fin de satisfacer las necesidades de los usuarios.

Como resultado de lo anterior, tanto los operadores como las PYMES (Pequeñas Y Medianas Empresas) proveedoras de aplicaciones, servicios y contenido, deben adecuar su infraestructura TI (Tecnologías de la Información) con el fin de desarrollar capacidades para crear y desplegar nuevos *servicios convergentes* en un corto tiempo de salida al mercado (*Time-to-Market*) [18]. Servicio convergente, se entiende como la coordinación de un conjunto de servicios de diferentes proveedores, tales que a la vista del usuario final sea un solo servicio [19], [20]. En otras palabras, las reformas mencionadas, permitirán a empresas del sector

que las acaten, adelantarse a la competencia en el lanzamiento de un servicio con el fin de generar altos beneficios para amortizar rápidamente los gastos de creación y despliegue.

Específicamente, la competitividad en las empresas del sector de las telecomunicaciones, va de la mano con la creación, adaptación, modificación e integración de servicios existentes; así como también, la adición de valor, considerando criterios óptimos de fiabilidad, calidad y reducción en el tiempo de despliegue. En este sentido, el objetivo es proveer, eficazmente, diversos tipos de servicios, independientes de la red de acceso del usuario final. Sin embargo, las prácticas habituales de los procesos de despliegue de servicios nuevos al interior de estas empresas, limitan la consecución de dichos objetivos, debido a tareas manuales o actividades que consumen una cantidad considerable de recursos y tiempo tales como: búsqueda, recuperación y composición de servicios. De esta forma, encontrar servicios que cumplan con los requerimientos/exigencias de los usuarios finales, se torna una tarea sumamente compleja; ya que, en contextos tan dinámicos y heterogéneos como los entornos convergentes, es imprescindible contar con capacidades óptimas de búsqueda de servicios, que permitan al usuario final explotar eficaz y eficientemente los recursos presentes en la red (Servicios de la Web y Telco).

En este contexto, uno de los retos claves de competitividad que deben contemplar las actuales empresas de telecomunicaciones, es la reducción del tiempo en el proceso de creación de nuevos servicios convergentes (Web 2.0 + Telco), agilizando su despliegue y puesta en el mercado. Por esta razón, recientemente se han generado proyectos como OPUCE (*Open Platform for User-Centric Creation and Execution*) [21], SPICE (*Service Platform for Innovative Communication Environment*) [22] o el motor de composición *Ericsson Composition Engine* (ECE) [23]. Dichos proyectos, han desarrollado entornos de creación de servicios convergentes, que requieren solicitudes formales con expresiones complejas por parte del desarrollador, para componer servicios que se seleccionan de manera manual. Esta recuperación manual de servicios, demuestra que el proceso de creación de servicios convergentes, depende de la habilidad del desarrollador para seleccionar los servicios más adecuados, lo que constituye una labor dispendiosa e ineficiente, puesto que se dificulta para la capacidad humana, generar composiciones manuales que vayan de acuerdo al crecimiento del número de servicios disponibles tanto en la Web como en el dominio de las telecomunicaciones.

En este orden de ideas, el tiempo de creación de servicios convergentes puede ser mejorado al disminuir la complejidad en las actividades realizadas por el *Prosumer*. Actividades como la selección de servicios atómicos que conforman un servicio compuesto, requieren ser más flexibles y ágiles [24-26]. Para conseguirlo, es necesario definir mecanismos que faciliten este proceso, como: capacidad para realizar peticiones en un lenguaje más simple y entendible, como el lenguaje natural; automatizar tareas en la identificación de requerimientos y selección de servicios, entre otros. De lo expuesto, el presente trabajo se centra en responder la siguiente pregunta de investigación:

**¿Cómo reducir la complejidad de la recuperación de servicios en un entorno convergente a usuarios finales?**

## **1.4. OBJETIVOS DE INVESTIGACIÓN**

### **1.4.1. Objetivo general**

Automatizar la recuperación de servicios en entornos convergentes considerando requerimientos funcionales y no funcionales, a partir de las solicitudes realizadas en lenguaje natural.

### **1.4.2. Objetivos específicos**

- Determinar los requerimientos funcionales y no funcionales de las solicitudes del usuario final realizadas en lenguaje natural.

- Identificar automáticamente servicios en un entorno convergente, considerando los requisitos funcionales y no funcionales de la solicitud del usuario, a partir técnicas y algoritmos de emparejamiento semántico.
- Desarrollar y evaluar experimentalmente un prototipo en un entorno convergente que permita visualizar los mecanismos propuestos.

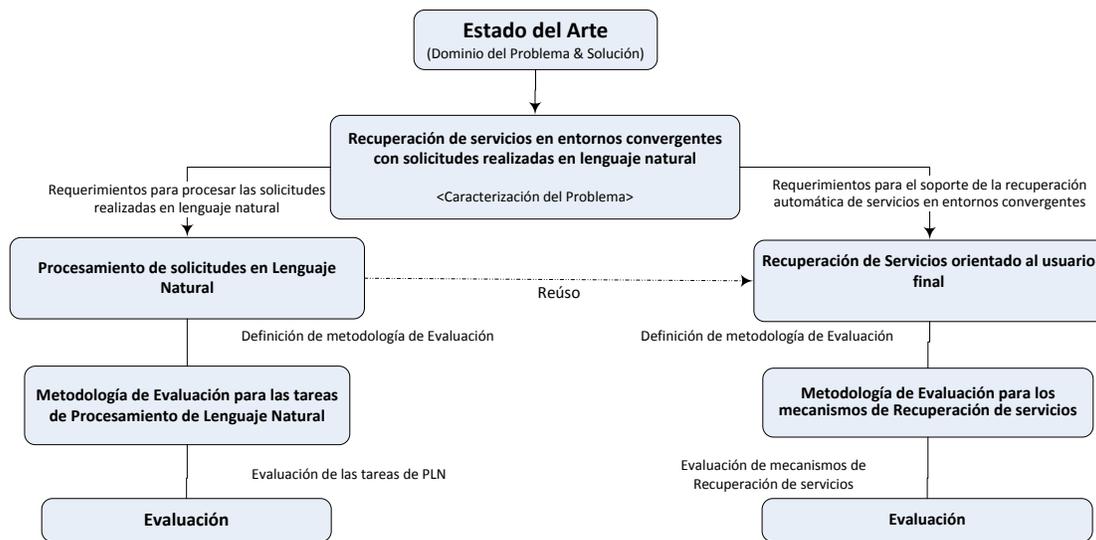
### 1.4.3. Hipótesis de Investigación

Como suposición lógica para el tema de investigación, se plantea una hipótesis y su hipótesis nula (**H1** y **H1<sub>0</sub>**), al demostrar que la hipótesis nula no es verdadera, se acepta la hipótesis de investigación **H1** [28]. Estas conjeturas son descritas a continuación.

- **H1:** *Es posible recuperar servicios en entornos convergentes a través de la captura de los requerimientos funcionales de la solicitud (NL) del usuario; sin embargo, los requerimientos no funcionales reducen la ambigüedad en el proceso de selección, permitiendo recomendar servicios más relevantes a la consulta del usuario.*
- **H1<sub>0</sub>:** *NO es posible recuperar servicios en entornos convergentes a través de la captura de los requerimientos funcionales de la solicitud (NL) del usuario; Los requerimientos no funcionales **no reducen** la ambigüedad en el proceso de selección, afectando la recomendación de servicios relevantes a la consulta del usuario.*

## 1.5. ENFOQUE

Con el fin de dirigir claramente los objetivos definidos para el presente trabajo de investigación, se ha usado el enfoque expuesto en la Figura 2.



**Figura 2. Enfoque de Investigación del Trabajo de Grado**

A partir de la revisión del estado del arte alrededor del tema de investigación planteado, es posible notar que la mayoría de técnicas de recuperación de servicios orientadas al usuario final trabajan sobre servicios en la Web, limitando el uso de otro tipo de recursos que pueden ser de interés para usuario, tales como los

servicios de telecomunicaciones. De igual manera, es posible observar que dichas aproximaciones se enfocan en el uso de propiedades funcionales (como parámetros operacionales: Entradas, Salidas, entre otros), restringiendo las características no funcionales que puedan tener los servicios (Por ejemplo, QoS). Otro de los aspectos negativos relevantes, es la definición de los requerimientos por parte del usuario en el proceso de recuperación, ya que, o representan una complejidad alta para los usuarios al tener que ser formulados en lenguajes formales, incomprensibles para personas sin conocimientos técnicos, ó son demasiado simples, limitándose a solicitudes con palabras clave ó frases que deben cumplir obligatoriamente con restricciones de lenguaje (uso de plantillas en la solicitud).

Para soportar la recuperación de servicios y permitir superar limitaciones en la especificación de los requerimientos de servicios por parte del usuario final, se ha propuesto un mecanismo que soporta la especificación de la petición del usuario en solicitudes realizadas en lenguaje natural. En el desarrollo de esta propuesta, se obtuvo como resultado un sistema de procesamiento de lenguaje natural compuesto por un conjunto de tareas, ejecutadas secuencialmente de manera automática, para la extracción de diferentes tipos de información. Para ello, se utilizaron ontologías léxicas, reconocimiento de expresiones regulares en documentos, anotaciones gramaticales y herramientas de autocompletado y autocorrección de texto; con el fin de brindar mayor soporte a los usuarios.

Por otro lado, conscientes de las limitaciones en procesos de recuperación que solo se enfocan sobre servicios Web, propiedades funcionales, o resultan excesivamente manuales y complejos para los usuarios, se adoptaron en una serie de enfoques necesarios para superar las restricciones nombradas, como: *i)* El uso de etiquetado colaborativo de términos, que permite anotar los diferentes recursos (Web y Telco) y a su vez, el usuario puede ver como otros han usado anotaciones en sus solicitudes, siendo posible usar dichas anotaciones para los servicios requeridos, simplificando el proceso desde el punto de vista del usuario. *ii)* Como complemento a lo anterior, el sistema sigue un modelo de recomendación donde al finalizar una solicitud, se almacena la salida del sistema con los términos usados por el usuario, de tal manera, para solicitudes futuras, esta salida pueda ser reutilizada como recomendación. *iii)* Definición de un proceso de expansión de los términos provenientes de la solicitud del usuario, a través de ontologías léxicas y fuentes de conocimiento social. *iv)* Para brindar una mayor consistencia al proceso de emparejamiento con propiedades no funcionales, fueron introducidos dos mecanismos diferentes, el primero considera todo el repositorio de servicios y los términos provenientes del usuario para la generación de un ranking preliminar de servicios; con base en este ranking inicial, el segundo mecanismo selecciona los primeros servicios y los reordena en un nuevo ranking, basándose en los valores de sus propiedades no funcionales.

De igual forma, como parte fundamental del enfoque propuesto, se encuentra la evaluación del prototipo desarrollado. Donde, debido a la ausencia de un sistema de procesamiento de lenguaje natural similar al propuesto, orientado a la identificación y expansión de términos de las solicitudes realizadas en lenguaje natural para la selección de servicios, fueron definidos dos tipos de metodologías de evaluación, Estas metodologías permiten realizar comparaciones objetivas, adaptadas al campo de investigación, y a su vez son útiles para evaluar tanto al sistema en general como mecanismos específicos que lo soportan. Adicionalmente, la evaluación del proceso de emparejamiento fue llevada a cabo con medidas más precisas y objetivas que las tradicionales, utilizadas en los sistemas de recuperación de servicios y de información. Los resultados de evaluación obtenidos fueron comparados con sistemas de recuperación similares.

En síntesis, lo anterior, similar a un mapa, define un esquema flexible y simple, que permite dirigir el cumplimiento de los objetivos de investigación planteados, con el fin de soportar la recuperación de servicios sobre entornos convergentes, considerando propiedades funcionales y no funcionales de los servicios, sobre un ambiente centrado en el usuario.

## 1.6. MODELO CONCEPTUAL

En este apartado se describe el modelo conceptual que representa el enfoque propuesto, con el fin de brindar una visión general del trabajo realizado y describir brevemente los componentes más importantes que lo soportan, así como su relación; tal como se presenta en la Figura 3.

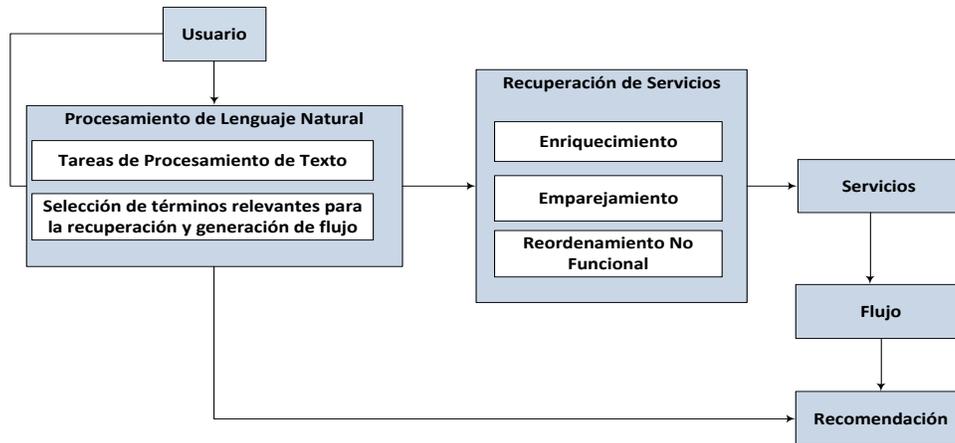


Figura 3. Modelo Conceptual Básico

El usuario solicita un conjunto de servicios a través de lenguaje natural, indicando los requerimientos necesarios. En el sistema, se realiza el procesamiento correspondiente de la solicitud, identificando los términos que pueden describir un servicio (nombres propios, dispositivos, lugares, verbos, adjetivos, etc.) o que sean relevantes para la generación de un flujo de servicios (comas, puntos, condicionales, conjunciones, etc.). A partir de los términos seleccionados que pueden describir un servicio, se adicionan algunos términos relacionados (Enriquecimiento), con el fin de expandir el conjunto de términos inicial obtenido de la solicitud del usuario.

Los términos seleccionados como relevantes para la generación del flujo de servicios, dividen la solicitud del usuario en bloques, donde cada bloque contiene un subconjunto de los términos expandidos, obtenidos tras el proceso previo de enriquecimiento. De esta manera, cada bloque corresponde a un proceso de emparejamiento diferente, cuyo fin es retornar la lista de todos los servicios disponibles, los cuales se encuentran ordenados según la relevancia respecto al subconjunto de términos expandidos que hace parte de dicho bloque.

En el proceso de reordenamiento, los primeros servicios del ranking del emparejamiento, son reorganizados considerando los valores de las propiedades no funcionales que contiene cada servicio, de acuerdo a preferencias elegidas por el usuario. De esta manera, se obtiene una nueva lista de servicios (una lista por bloque). Con las listas de servicios y los términos obtenidos para el flujo, el sistema construye un flujo de servicios, usado para indicar el orden de los mismos y desplegar los resultados al usuario.

Finalmente, el flujo junto con los términos del usuario utilizados para el proceso de emparejamiento, son almacenados para ser usados como recomendación para futuras solicitudes del mismo u otros usuarios. Estas solicitudes son mostradas mientras se ejecuta todo el proceso descrito, el cual es detenido en caso de que el usuario prefiera alguna recomendación, utilizando la salida registrada anteriormente.

## 1.7. ALCANCE

En el presente proyecto, se analizaron algunas tareas para el procesamiento de lenguaje natural, con el fin de extraer términos que expusieran la descripción de un servicio en términos de parámetros funcionales y no funcionales. Este proyecto abarca el ingreso de texto hablado (reconocimiento de voz) y escrito en

inglés. Por otro lado, se consideraron solicitudes informales, con un grado de coherencia aceptable; es decir, solicitudes con palabras en otro idioma o palabras sin significado como: “xyasewrf”, solo despliegan un error subrayando dichas palabras. Aun así, tales términos serían considerados para pasar por los diferentes mecanismos de procesamiento como si fueran otra palabra común.

Para la fase de emparejamiento (ver Figura 3), en caso de especificarse en la solicitud, valores numéricos de criterios no funcionales, estos no se consideraron; es decir, para solicitudes como: “Quiero un servicio de llamada con disponibilidad del 99.999%”, el sistema solo considerará la búsqueda de servicios relacionados con llamada y disponibilidad, haciendo caso omiso al valor 99.999%. Sin embargo, para la fase de reordenamiento no funcional (ver Figura 3), si se considera los valores específicos de las propiedades no funcionales de cada servicio en el ranking generado. Estas propiedades fueron proporcionadas por diversas fuentes en la Web, donde se obtuvieron servicios y un total de 9 propiedades no funcionales; sin embargo, el sistema permite extender el número de propiedades según sea requerido.

Finalmente, el presente proyecto, solo considera la descripción de los recursos (servicios Web y Telco) por medio de términos anotados (etiquetas) que describen su funcionalidad, calidad y entradas o salidas, a través de archivos que extiendan las descripciones de la interfaz de los servicios (WSDL).

## 1.8. CONTRIBUCIONES

Las principales contribuciones de éste proyecto de grado son:

- Este trabajo es un aporte al módulo de recuperación de servicios dentro del proyecto TelComp 2.0 (Recuperación y Composición de Componentes Complejos para la Creación de Servicios Telco 2.0) llevado a cabo en la Universidad del Cauca y financiado por COLCIENCIAS. Proyecto Código: 1103-521-28338, Contrato RC 458-2011.
- **Definición y aplicación de un mecanismo de Procesamiento de Lenguaje Natural (PLN).** Considerando el dominio de conocimiento sobre el cual se centra el proyecto que se documenta en esta monografía, se llevó a cabo una exploración de sistemas de PLN y tareas en los diferentes niveles lingüísticos. A partir de esta exploración, se determinó emplear un conjunto de 6 tareas en cascada: tokenización, filtrado de palabras, etiquetado de palabras, clasificación de nombres de entidad, desambiguación semántica y lematización. Estas tareas se adaptaron bajo una misma plataforma de ejecución que permite definir las dentro de un flujo de tareas en cascada, donde la salida de una corresponde a la entrada de la otra.
- **Selección y adaptación de un mecanismo de enriquecimiento de vocabulario.** A partir de los términos obtenidos de la solicitud en lenguaje natural, se estableció un mecanismo que permite expandir dichos términos para solucionar problemas atribuidos al uso de vocabulario de lenguaje natural (sinónimos, términos ambiguos, uso de abreviaturas, entre otros) y la falta de información en la descripción de los servicios. Para ello se consideraron dos fuentes de conocimiento: por un lado, una fuente de conocimiento formal, como lo es una ontología léxica y por otro, una fuente de origen social, que permitiera complementar a la primera en la expansión de términos. Esto conlleva al siguiente aporte.
- **Selección y adaptación de dos fuentes de conocimiento.** Tras la exploración de diferentes tipos de representación de conocimiento, se eligió el uso de una fuente formal como WordNet que permita establecer relaciones semánticas y léxicas para enriquecer el vocabulario recuperado de las solicitudes realizadas por los usuarios. Para ello, fueron definidas diferentes relaciones entre términos, considerando aquellas que WordNet provee: hiperónimos, hipónimos, holónimos, merónimos, etc. Adicionalmente, se determinó emplear Flickr, como fuente social, con el fin de usar sus clústeres de etiquetas. Cabe resaltar que en un clúster los términos presentan una relación de carácter social, dada por los usuarios de esta plataforma; es decir, cada clúster posee términos dados por usuarios que

desde su perspectiva están relacionados (P. ej. mascota-labrador están relacionados). Esto como complemento para WordNet.

- **Definición de un mecanismo de recuperación automática de servicios.** Con la revisión de diferentes soluciones sobre recuperación y búsqueda de servicios, fue posible proponer un enfoque, que permite adaptar la recuperación de servicios Web a cualquier tipo de recurso consumible (Widgets, APIs, aplicaciones, etc.), tornándolo adecuado a las características y requisitos propios de los entornos convergentes. Además, puede configurarse fácilmente, para volver el proceso de recuperación automático o también con una controlada intervención del usuario (semiautomático).
- **Adaptación de un mecanismo de recuperación de servicios con parámetros no funcionales.** Un mecanismo de recuperación de servicios que consideraba parámetros no funcionales, fue adaptado para cumplir con los intereses del presente proyecto. Así, los algoritmos del mecanismo son utilizados para la fase de reordenamiento no funcional (ver Figura 3), siendo ejecutado como un subproceso dentro del sistema desarrollado.
- **Definición de un repositorio de servicios para el proceso de recuperación** Este aporte corresponde a un modelo de base de datos relacional, el cual permite establecer una correspondencia entre las propiedades no funcionales y sus valores respectivos, con cada uno de los servicios que se desean ofrecer. Adicionalmente, son definidos 4 tipos de descripción funcional de un servicio: los términos que describen la entrada/salida (sustantivos, adjetivos o propiedades no funcionales) del servicio, las operaciones/características (comportamiento) de un servicio; estos a su vez se clasifican dependiendo, si son términos provenientes de la solicitud del usuario o si son enriquecidos semánticamente.
- **Prototipo de recuperación automática de servicios en un entorno convergente a partir de solicitudes realizadas en lenguaje natural.** Se desarrolló un prototipo, el cual implementa todos los aportes anteriormente propuestos. Esta herramienta permite la ejecución de los diferentes algoritmos definidos que intervienen en el emparejamiento, y la recuperación de servicios sobre entornos convergentes. La plataforma desarrollada se centra en dos tipos de usuarios: Los desarrolladores y usuario finales, generando así una aplicación de escritorio y móvil respectivamente.
- Se destaca la producción de un artículo presentado en “*The Third International Conference on Advanced Service Computation – SERVCOMP 2011*”, llevada a cabo entre los días 25-30 de Septiembre del 2011 en Roma, titulado: “*Automatic Service Retrieval in Converged Environments Based on Natural Language Request*”. En esta contribución se expone de manera general el mecanismo de PLN propuesto, así como la arquitectura genérica para la recuperación de servicios en entornos convergentes centrada en el usuario. Este artículo fue galardonado como “*Awarded Best Paper*” o “Mejor Artículo”, además fue invitado a extender su contenido en el *International Journal On Advances in Software* (vol 5 nr 1&2 del año 2012).
- Se destaca la producción de un artículo expuesto en “*17th International conference on Applications of Natural Language Processing to Information Systems – NLDB 2012*”, llevada a cabo entre los días 26-28 de Junio del 2012 en Groningen, Holanda. Este artículo se titula: “*User-Driven Automatic Resource Retrieval Based on Natural Language Request*”. En esta contribución se describe y evalúa experimentalmente el mecanismo de PLN. Para esto, en primera instancia se propone una definición formal de las diferentes entidades que intervienen en el procesamiento de la solicitud del usuario. Este artículo fue publicado en una importante serie de los libros de LNCS (*Lecture Notes in Computer Science*) por Springer-Verlag Berlin Heidelberg 2012.
- Se destaca la producción de un artículo publicado en la revista internacional “*International Journal on Advances in Software*”, vol 5 del 2012, titulado “*User-driven Service Retrieval Platform for Converged Environments*”, en el cual se expone los resultados sobre la satisfacción de la salida del sistema

desarrollado, producto de una evaluación realizada por expertos en el tema de recuperación y composición. Asimismo, se amplía la descripción funcional de todos los componentes señalados en el artículo “User-Driven Automatic Resource Retrieval Based on Natural Language Request”.

- Motivar el desarrollo y la investigación sobre los servicios convergentes en la comunidad académica de la Universidad del Cauca, implementando un prototipo de recuperación de servicios sobre entornos convergentes que cree un precedente alrededor de este tema, y despierte así el interés por estas tecnologías en la comunidad científica del país, debido al factible impacto (social y económico) positivo, obtenido a través de su difusión.

## **1.9. CONTENIDO DE LA MONOGRAFÍA**

### **Capítulo 2. Estado del Arte**

En este capítulo se abordan las definiciones formales de conceptos claves para el entendimiento del proyecto, tales como: tareas de Procesamiento de Lenguaje Natural, Sistemas de Representación de Conocimiento, Convergencia; además del estado actual del conocimiento relacionado con trabajos que se han adelantado sobre las diferentes áreas que abarca este proyecto.

### **Capítulo 3. Análisis del Lenguaje Natural**

En este capítulo se hace una descripción del análisis de lenguaje natural y se definen sus respectivos componentes, justificando la selección y adaptación de los componentes necesarios para el desarrollo del presente proyecto.

### **Capítulo 4. Recuperación de Servicios Orientado al Usuario Final**

En el capítulo se presenta el estudio y selección de los mecanismos para la elaboración de un ranking de servicios atómicos, considerando los términos obtenidos de la solicitud del usuario. Igualmente, se describe el proceso de refinamiento del ranking generado, con base en las propiedades no funcionales, con el fin de obtener servicios más precisos y reducir el espacio de búsqueda

### **Capítulo 5. Prototipo y Evaluación**

En este capítulo se aborda la descripción detallada del trabajo realizado alrededor del desarrollo de un sistema software, el cual implementa la propuesta del proyecto. En la descripción se abarca la arquitectura definida para soportar la construcción del prototipo, los modelos de caso de uso del sistema, detalles de implementación del prototipo y la metodología de evaluación del mismo, junto con los resultados obtenidos.

### **Capítulo 6. Conclusiones y Trabajo Futuro**

Por último, se analizan los resultados obtenidos, detallando las principales contribuciones de la ejecución del proyecto y brindando un conjunto de recomendaciones importantes para el desarrollo de trabajos futuros.

## 2. ESTADO DEL ARTE

En este capítulo son presentados las bases conceptuales, definiciones formales y el estado actual de los trabajos que abarcan temas relacionados con este proyecto. Inicialmente, se describen los conceptos claves en los que se fundamenta la propuesta de recuperación de servicios a partir de lenguaje natural sobre entornos convergentes. Posteriormente, se profundiza en los temas principales de investigación: Recuperación de Servicios y Procesamiento de Lenguaje Natural, esto con el fin de resaltar los avances y las brechas existentes en dichos campos. Finalmente, se presenta un resumen describiendo los aportes más importantes del capítulo.

### 2.1. CONCEPTOS GENERALES

#### 2.1.1. Entorno Convergente

Con la digitalización de contenido y el cambio a redes basadas en IP, el acceso a ancho de banda de alta velocidad y la disponibilidad de comunicaciones multimedia y dispositivos de computación, se ha logrado que arquitecturas de Internet y telecomunicaciones evolucionen constantemente hacia una sola red, tanto en capas físicas y de transporte, como en la capa de servicio [1]. Lo anterior se evidencia en la aparición de entornos convergentes, en los cuales es posible ofrecer un diverso conjunto de servicios, independiente de la red de acceso que el cliente posea [1]. La convergencia se lleva a cabo en diferentes niveles:

- *Convergencia de Red*: soportada por el cambio a redes basadas en IP, constituye la convergencia de la red fija, móvil y de redes de conexión y difusión (p.ej. Triple Play) [42].
- *Convergencia de Servicios*: producto de la convergencia de red, permite el acceso a aplicaciones basadas en la Web, y el consumo de servicios tradicionales de telecomunicaciones, así como la generación de nuevos servicios de valor agregado, independientemente del canal de comunicación usado por el usuario [42].
- *Convergencia de Industria/Mercado*: hace referencia a la convergencia de las industrias de tecnologías de información, telecomunicaciones y medios que operan en mercados separados [42].
- *Convergencia legislativa, institucional y regulatoria*: se refiere a la regulación de contenido y servicios independientemente de la red sobre la cual se proveen [42].
- *Convergencia de Dispositivos*: se refiere a las capacidades y altas prestaciones de los dispositivos móviles actuales para proveer múltiples funciones de comunicación y acceso a aplicaciones [42].
- *Convergencia de experiencia del usuario*: provisión de una interfaz única entre usuarios finales y tecnologías de telecomunicaciones, medios e informática.

Considerando los niveles de convergencia expuestos, el interés principal para el desarrollo del presente proyecto se centra en la convergencia de servicios y de experiencia del usuario. Para esto, se ha considerado el despliegue de un entorno convergente (de prueba) sobre Internet, donde es posible encontrar recursos disponibles, tales como servicios Web y/o capacidades de telecomunicaciones. Este contexto de prueba cuenta con un entorno de ejecución de lógica de servicios, usado para ejecutar las capacidades de telecomunicaciones, las cuales a su vez, pueden invocar servicios en la Web.

#### 2.1.2. Recuperación de Servicios en la Web

La recuperación de servicios en la Web puede verse desde dos perspectivas: sintáctico y semántico. La búsqueda de servicios en la Web desde el enfoque sintáctico, considera técnicas de comparación de interfaces (por ejemplo: WSDL, IDL, entre otros) o búsquedas por palabras claves [1], [41]. Estas técnicas requieren coincidencias exactas a nivel sintáctico entre las descripciones de los servicios y la solicitud realizada. Por tal razón, este tipo de búsqueda obtiene resultados de baja calidad, ya que no considera las relaciones existentes entre las palabras de la petición realizada [42], impidiendo la captura de la semántica

de la solicitud y obteniendo en ocasiones servicios ambiguos que no corresponden al criterio de búsqueda del usuario.

A su vez, el enfoque semántico permite el establecimiento de relaciones entre conceptos que definen la funcionalidad de los servicios, es decir, sus propiedades funcionales (p.ej., entradas, salidas, precondiciones y efectos de un servicio) y no funcionales (p.ej., disponibilidad, tiempo de respuesta, reputación de un servicio, entre otros) [43]. Por tanto, este enfoque permite una mayor precisión en la descripción de los servicios, mejorando la calidad de los resultados al recuperar servicios más acordes a las necesidades del usuario [9], [44]. Esta característica es importante, ya que permite soportar la recuperación automática de servicios [46].

### 2.1.3. Recuperación de Servicios de Telecomunicaciones

En cuanto a la recuperación de servicios en el dominio de las telecomunicaciones, se puede afirmar que corresponde a un proceso con una notable diferencia respecto a la búsqueda de servicios en la Web. En la actualidad, debido a la complejidad de este proceso, diversas alternativas han sido desarrolladas con el fin de reducirla y así promover una recuperación más simple y eficiente. De esta manera, surgen proyectos importantes como SIMS [47] (*Semantic Interfaces for Mobile Services*), SPICE (*Service Platform for Innovative Communication Environment*) [21]. SIMS permite capturar la semántica de las características funcionales y no funcionales de los servicios del mundo Telco, haciendo posible realizar comparaciones aproximadas entre la solicitud y los servicios disponibles, para la recuperación deseada y precisa de servicios de telecomunicaciones [49]. SPICE, por su parte ofrece un entorno de creación de servicios, que facilita la recuperación de capacidades de telecomunicaciones, las cuales han sido descritas con diferentes tipos de anotaciones semánticas; además, dicho entorno de creación, provee adaptadores que soportan la ejecución de servicio en diferentes plataformas de ejecución como JainSlee (JSLEE) [145].

### 2.1.4. Representación de Conocimiento

#### 2.1.4.1. Folcsonomías

Derivado de *Taxonomía*, el término *Folksonomy* ha sido atribuido a Thomas Vander Wal. Taxonomía procede del griego "*taxis*" y "*nomos*": *Taxis* significa clasificación y *nomos* (o *nomía*), ordenar, gestionar. Por su parte, "*folc*" proviene del alemán "*pueblo*" (*volks*). En consecuencia, de acuerdo con su formación etimológica, *folcsonomía* (*folc+taxo+nomía*) significa literalmente "clasificación gestionada por el pueblo".

La idea es muy simple: cuando un usuario gestiona un recurso (imagen, video, etc.) en un sitio, por lo general público, lo hace asignándole varias palabras claves, siempre bajo su criterio personal. De esta forma, el proceso de etiquetar (*tagging* en inglés) provee los mecanismos necesarios para que usuarios creen anotaciones o metadatos con palabras claves conocidas como *tags* [33] asociadas a recursos, con el fin de compartirlas, describirlos, indexarlos y/o categorizarlos [34]. Así, cuando en estos sistemas de etiquetado es posible que más de un usuario gestione los metadatos realizados sobre los recursos, se habla entonces de etiquetado colaborativo [33], [34] o Folcsonomías, es decir: una taxonomía hecha por la gente [40].

Actualmente, las folcsonomías son ampliamente utilizadas en aplicaciones de etiquetado de la Web Social. Una folcsonomía permite, de manera transparente y colaborativa, anotar y categorizar recursos por etiquetas (palabras claves). Las folcsonomías reflejan así, a través de etiquetas, la *Inteligencia Colectiva* de un grupo de personas (*Wisdom of the crowds* - Sabiduría de las multitudes) que da significado a un conjunto de recursos disponibles [76].

Entre las ventajas y fortalezas más sobresalientes de las folcsonomías se encuentran, la creación y clasificación de contenidos sociales, flexibles, dinámicos y dependientes del usuario. Se denomina "dependientes" porque las anotaciones realizadas a los recursos representan el conocimiento e interés del

usuario que realizó el etiquetado, reflejando el vocabulario usado por el mismo. Esto a su vez, permite una gran adaptabilidad de los sistemas, convirtiéndolos en sistemas altamente accesibles frente a usuarios que no tienen destrezas técnicas o conocimientos profundos sobre los recursos.

Sin embargo, este tipo de sistemas presentan algunos inconvenientes como: la ambigüedad en el significado (polisemia al usar la misma etiqueta en varios recursos correspondientes a diferentes contextos), variación de etiquetas (sinónimos), falta de organización de etiquetas, errores tipográficos, uso de espacios o puntuación en las etiquetas [34], entre otros. Estos problemas conllevan a una baja precisión en los resultados cuando se ejecutan búsquedas sobre los recursos.

#### **2.1.4.2. Ontologías**

Si bien no existe una definición estándar de lo que es una ontología, la definición de la W3C (World Wide Web Consortium) ha sido ampliamente aceptada: *“una ontología define los términos utilizados para describir y representar un área de conocimiento”* [77]. Como complemento a esta definición, se tiene la propuesta en [35]: la cual menciona que: *“una ontología es la especificación formal y explícita de una conceptualización compartida”*. En esta definición, *conceptualización* hace referencia a un modelo abstracto de algún fenómeno del mundo que representa sus conceptos más relevantes; *explícito* se refiere a la necesidad de especificar de forma consciente los distintos conceptos que conforman una ontología; *formal* indica que la especificación debe representarse por medio de un lenguaje de representación formalizado; y finalmente, *compartida* refleja una característica relevante, que una ontología debe, en el mejor de los casos, dar cuenta de un conocimiento aceptado [36]. Según estas definiciones, es posible afirmar que las ontologías representan un vocabulario unificado que describe el conocimiento sobre el mundo o parte de él, que define los conceptos y relaciones para dicho conocimiento.

Por su parte, las ontologías permiten desarrollar sistemas con modelos de conocimiento caracterizados por ser los más formales, sólidos y estáticos. Sin embargo, también son las más costosas en términos de creación y mantenimiento, ya que requieren un acuerdo del contenido con los miembros que la van a usar y generalmente, los usuarios de dichas ontologías no están involucrados en su proceso de desarrollo [37]; esto se debe porque son diseñadas por expertos y no usuarios finales, a diferencia de las folcsonomías.

#### **2.1.4.3. Taxonomías**

El término taxonomía proviene del griego, *“Taxis”*, que significa *clasificación* y *“nomos”*, que indica ciencia [38] para ordenar y/o gestionar. Por tanto, estos sistemas de clasificación permiten convertir información sin estructura en conocimiento organizado. De manera más amplia, se puede pensar el concepto de taxonomía como un medio para organizar conceptos de conocimiento [39], pero en un sentido más preciso, significa una clasificación jerárquica o sistema de categorización. Aún con la abstinencia por parte de profesionales para usar este término, ya que es demasiado ambiguo y usado equívocamente, ha ganado suficiente popularidad y parece no existir en la actualidad un término alternativo que lo remplace. El concepto de taxonomías como sistemas de clasificación jerárquica, se ajusta actualmente a la organización de recursos, especialmente en el dominio Web. Una definición más profunda de este concepto es expuesta en [39], donde una taxonomía es *“una clase de vocabulario controlado, en el cual cada término es conectado/relacionado a un término más genérico (-a menos que este sea el término más alto de la jerarquía) y a uno o más términos más profundos (a menos que este sea el término del último nivel), siendo todos estos términos organizados en una estructura jerárquica única”*. En la definición, cuando se habla de vocabulario controlado, se refiere a una lista restringida de palabras o términos para algún propósito, sea indexar, etiquetar (*tagging*) o categorizar. Asimismo, existe un control sobre quién, cómo y cuándo se puede agregar términos a la lista; a diferencia de las folcsonomías, las cuales permiten que cualquier individuo, diferente al creador del contenido, agregue términos.

El beneficio clave de representar conocimiento con taxonomías, es el de agrupar términos relacionados y categorizarlos de manera que sea más fácil ejecutar búsquedas y encontrar los términos correctos. Aunque

las taxonomías son ampliamente usadas en muchos sitios Web y en directorios de navegación para la recuperación de contenido/conocimiento [40], se ven claras restricciones debido a la carencia para capturar aspectos culturales y de vocabulario de los usuarios. Esto se debe a que las taxonomías fueron diseñadas y/o creadas por una autoridad propietaria y no por usuarios finales. En este sentido, al igual que las ontologías, construir y mantener una taxonomía es una tarea que consume considerables recursos [41], sin mencionar su incapacidad de actualización cuando nuevos conceptos surgen.

Hay ciertas ventajas y desventajas en cada uno de las representaciones de conocimiento mencionadas. Integrando los tres tipos de representaciones expuestos en un único modelo, surgen características importantes que mejoran notablemente la búsqueda, tal como se puede observar en la Tabla 1. Por este motivo, el presente proyecto considera la integración de las representaciones descritas, con el fin de superar sus limitaciones por separado. Esta propuesta se detalla con mayor profundidad en la sección 4.

**Tabla 1. Modelo Híbrido de Representación de Conocimiento, adaptado de [78]**

Taxonomía	Folksonomía	Ontología	Modelo Híbrido
Control central	Creación democrática	Creación por expertos	Control central con entrada continua de usuarios
Preciso	Medianamente preciso	Preciso	Precisión validada por la comunidad
Vocabulario Definido	Vocabulario Personal	Vocabulario Técnico definido	Vocabulario de una comunidad refinado por vocabulario definido
Términos ambiguos	Términos ambiguos	Reducción de la ambigüedad de términos	Términos de la comunidad con menor ambigüedad
No dinámico	Dinámico y adaptable	No dinámico	Dinámico y adaptable
Organización estructurada	Poco organizada	Organización mejor estructurada	Modelo flexible y organizado

### 2.1.5. Procesamiento de Lenguaje Natural

Por lo general, el conocimiento humano es expresado y representado por medio de lenguajes (p.ej. lenguaje oral, escrito, por gestos, etc.). Con las tecnologías enfocadas en estos lenguajes, ha sido posible facilitar la comunicación *hombre-máquina*, ofreciendo interfaces y sistemas de acceso a información cada vez más sofisticados, por lo cual, en la actualidad, el Procesamiento de Lenguaje Natural (PLN) se ha convertido en una pieza fundamental para el desarrollo de la sociedad de información.

Actualmente no existe una definición estándar de lo que es el PLN, sin embargo una aproximación muy aceptada es la definición dada en [29], según la cual, “*PLN es un conjunto de técnicas computacionales utilizadas para analizar y representar naturalmente textos, en uno o más niveles de análisis lingüísticos, con el propósito de lograr el procesamiento de lenguaje humano para un rango de tareas o aplicaciones*”. Según esta definición, el desarrollo de sistemas PLN tiene como propósito procesar textos de cualquier tipo, sea hablado o escrito de la misma forma como lo hacen los humanos, para que las máquinas puedan entenderlo. Igualmente, en la definición se mencionan “niveles de análisis lingüísticos (ver sección 3.1.1)”, haciendo referencia a los múltiples tipos de procesamiento del lenguaje tanto para su producción y comprensión [15]. Generalmente, los sistemas de PLN, combinan varios de estos niveles de acuerdo a diversos propósitos, y a su vez, usan diferentes tipos de tareas de procesamiento, cada una con objetivos específicos sobre el texto de entrada en el sistema.

### **2.1.5.1. Niveles Lingüísticos**

La lingüística es la ciencia del lenguaje y cubre sonidos (fonología), formación de palabras (morfología), estructura de frases (sintaxis), significado (semántica) y entendimiento (pragmático) principalmente [19].

A continuación se describen los diferentes niveles de procesamiento de lenguaje, en los cuales la lingüística concluye que los humanos extraen significados:

- **Fonológico:** abarca la interpretación de los sonidos de las palabras, tiene en cuenta reglas prosódicas (intensidad con que se pronuncia una sílaba), reglas fonéticas para sonidos dentro de palabras y en conjunto.
- **Morfológico:** este nivel trata la naturaleza de las palabras, de acuerdo a sus morfemas, el sentido mismo de las palabras puede estar determinado por el uso adecuado de sufijos o prefijos dentro de su estructura.
- **Léxico:** identifica el significado de palabras individuales. Puede ser necesario el uso de un lexicón (diccionario que especifica, rasgos característicos de palabras, morfemas e información léxica), para identificar su verdadero sentido.
- **Sintáctico:** se centra en el análisis de las palabras en una oración, con el fin de descubrir la gramática o estructura de la oración. Su análisis, también requiere de gramática y su resultado corresponde a las dependencias o relaciones estructurales entre las palabras.
- **Semántico:** es el nivel más complejo, pues trata de determinar el posible significado que presenta una frase; sin embargo, frecuentemente se presentan problemas de ambigüedad, donde varios posibles sentidos pueden llegar a ser válidos en una oración [52], [53].
- **Discurso:** trabaja con unidades de texto más grandes que frases simples. Este nivel se enfoca en las propiedades de un texto y en el significado que transmite.
- **Pragmática:** este nivel va más allá de las reglas o estructuras gramaticales. Trata de comprender o explicar un significado adicional, entendido como intenciones, planes o metas, que no se encuentran implícitos en la estructura gramatical [54].

Es así como el PLN permite determinar el significado de un texto en los diferentes niveles; por lo cual, cabe aclarar, que no solo se habla de un nivel semántico cuando del significado de un texto se trata, como comúnmente se podría pensar, sino que se refiere a un conjunto de niveles, donde cada uno aporta un sentido distinto y es en conjunto donde es posible comprender expresiones lingüísticas de los seres humanos.

Tras la revisión de los diferentes niveles lingüísticos, el único nivel que no fue considerado fue el pragmático. La razón es porque se encuentra fuera del alcance del proyecto, el comprender el significado adicional de una solicitud (p.ej. si la intención con la que se solicitó un servicio es una pregunta ó una afirmación). El interés del proyecto, al abarcar los niveles lingüísticos, está fuertemente relacionado con las reglas ó estructuras gramaticales para la selección de términos importantes que pueden describir un servicio. La descripción detallada de las técnicas usadas de los diferentes niveles es ampliada en la sección 3.

### **2.1.5.2. Enfoques de PLN**

#### **Enfoque Simbólico**

Los enfoques simbólicos realizan un análisis profundo de los fenómenos lingüísticos, a través de esquemas de representación de conocimiento y algoritmos relacionados [30]. Un ejemplo de esto, son los basados en reglas, los cuales definen un conjunto de reglas, una máquina de inferencia y una memoria de trabajo, donde la máquina de inferencia recibe una entrada en lenguaje natural y selecciona repetidamente una regla cuya condición se cumpla para dicha entrada y la ejecuta [31].

### **Enfoque Estadístico**

En este enfoque se emplean varias técnicas probabilísticas para desarrollar modelos de fenómenos lingüísticos aproximados basados en ejemplos actuales de estos fenómenos [30]. Su importancia radica en la habilidad para obtener un buen desempeño, aún con conocimientos lingüísticos incompletos sobre un dominio de aplicación, frente al prerrequisito esencial de los enfoques simbólicos que requieren un completo entendimiento de dichos conocimientos.

### **Enfoque Conexionista**

En este enfoque son usadas redes neuronales que cumplen la función de almacenamiento de conocimiento en las diferentes conexiones. De esta forma, el concepto conexionista se puede entender como una red interconectada de unidades de procesamiento simples, con conocimiento almacenado en los pesos de las conexiones entre las unidades [30]. Así, una red neuronal es definida como un sistema de procesamiento de datos que consiste en un largo número de elementos de procesamiento simples (neuronas artificiales) altamente interconectadas en una arquitectura inspirada por la estructura cerebral de la corteza del cerebro [32].

### **Enfoque Híbrido**

Por su parte, esta aproximación explora diferentes variaciones de arquitecturas compuestas y modelos lingüísticos, con el objetivo de usar el mejor enfoque (simbólico, estadístico o conexionista) para un problema determinado [30].

Tras el estudio de los distintos enfoques sobre técnicas de procesamiento de lenguaje natural, la propuesta que más se ajusta al proyecto es la aproximación *híbrida*. Esto porque permite integrar las ventajas de los diferentes enfoques a fin de obtener mejores resultados y permitir una flexibilidad y escalabilidad en el uso de nuevos mecanismos de procesamiento. En la Sección 3 se expone una descripción detallada del enfoque usado.

## **2.2. TRABAJOS RELACIONADOS**

Como se describe en [15], tradicionalmente existen diversos enfoques orientados a la recuperación de servicios que consideran solicitudes formales y restringen la búsqueda principalmente a servicios Web, prescindiendo de otros recursos como las capacidades de Telecomunicaciones. Cabe resaltar, que en su mayoría, estos enfoques solo consideran parámetros funcionales y no proveen mecanismos que faciliten dicho proceso a usuarios inexpertos o con pocos conocimientos técnicos. En este sentido, esta sección se organiza con relación a las siguientes consideraciones:

Los enfoques relacionados con la recuperación de servicios son organizados con base en el tipo de servicio que se desea recuperar. Por fines prácticos, el presente trabajo se enfoca principalmente sobre servicios en la Web y de Telecomunicaciones. Dado a las características de la propuesta y las aproximaciones empleadas para la descripción semántica de los servicios, es posible extender los recursos de interés a: Widgets, API's, videos, fotos, entre otros recursos Social Media. De igual forma, son consideradas las facilidades que se ofrecen al usuario para simplificar el proceso de recuperación, así como también las propiedades no funcionales de los servicios dentro de este proceso.

Por otra parte, y como complemento a los enfoques que facilitan el proceso de recuperación de servicios a usuarios inexpertos, se lleva a cabo un estudio de trabajos relacionados con PLN, los cuales se organizan de acuerdo al nivel de complejidad de las tareas usadas para procesar solicitudes, así como la relación que tienen con sistemas de recuperación de servicios.

## **2.2.1. Recuperación de Servicios**

### **2.2.1.1. Recuperación de Servicios en la Web**

En este grupo, se describen las propuestas de recuperación de servicios Web, dentro del cual se pueden identificar dos enfoques de acuerdo a si están orientados al usuario final o no.

#### ***Recuperación de Servicios Web orientada al Usuario final***

Debido a las limitadas habilidades técnicas, que por lo general, los usuarios finales presentan, los enfoques en este grupo poseen características particulares, con las cuales buscan brindar el soporte necesario para facilitar el proceso de recuperación de servicios Web. A continuación, se realiza una clasificación de las propuestas más relevantes de acuerdo a dichas facilidades: recuperación con solicitudes en lenguajes informales (lenguaje natural), expansión de términos de la solicitud y formas de sugerencia.

Inicialmente la recuperación con solicitudes en lenguajes informales, abarca el uso de términos comprensibles por el usuario, normalmente usando el mismo idioma con el que se comunica. En esta categoría se describen enfoques que usan solicitudes expresadas en lenguajes controlados (frases restringidas a estructuras predefinidas) [4], [55], propuestas que utilizan como entrada palabras claves o etiquetas [57], [115] y finalmente sistemas que permiten el uso de lenguaje natural informal [9,48].

Otra facilidad es la expansión de términos de una solicitud [68], esta permite agregar nuevos términos a la petición del usuario, con el fin de complementarla y lograr una recuperación más relevante. Típicamente este proceso es realizado usando fuentes de conocimiento como ontologías, folcsonomías o taxonomías. En el caso de [34], el sistema agrega términos de ontologías y clústeres de etiquetas de sitios Web como Flickr.com para anotar y recuperar videos previamente importados de Youtube.com. Un proceso muy parecido es seguido en [58], donde se realiza un proceso de expansión semántica, usando ontologías en línea indexadas en Watson (Watson Semantic Web Gateway) [59] y WordNet [79].

Como otro de los soportes importantes para el proceso de recuperación de servicios Web, los mecanismos de recomendación, permiten acompañar al usuario en el proceso, ofreciendo servicios de acuerdo a solicitudes previas similares u otro tipo de ayudas, las cuales permiten realimentar al usuario y facilitar la selección de servicios adecuados. De la revisión del estado del arte, se identificaron 2 enfoques diferentes. El primer enfoque se centra en la realización de recomendaciones léxicas para la solicitud del servicio y el segundo, en realizar recomendaciones de servicios frente a la solicitud. Hacen parte del primero, trabajos como [33], el cual gira en torno al etiquetado colaborativo de servicios Web, de esta manera, los usuarios pueden anotar servicios Web con palabras clave y a la vez consultarlos. Para impulsar a la comunidad a ser miembros activos y realizar anotaciones a servicios constantemente, los usuarios pueden ver etiquetas realizadas por la comunidad en general. Así, los términos usados en una consulta para solicitar un servicio, pueden contener términos previamente utilizados por otros usuarios.

El otro enfoque permite que el usuario visualice listas de servicios a partir de su solicitud, antes de que se realice el proceso de recuperación. Un ejemplo de este enfoque es el sistema descrito en [60], el cual tras recuperar los verbos de la solicitud del usuario, identifica los nombres de las interfaces de servicio y genera una lista con los servicios candidatos, los cuales pueden servir como recomendación. Posteriormente, con esta lista, el sistema calcula la similitud entre parámetros de cada servicio con los nombres y adjetivos para el retorno del servicio más apropiado.

#### ***Recuperación de Servicios Web orientada a Expertos***

La recuperación de servicios Web centrada en expertos, reúne los trabajos y sistemas tradicionales, que reciben como entrada lenguajes formales (SPARQL, OWL, RDF entre otros) [67], [68], [69], y no soportan un

acompañamiento al usuario durante el proceso, siendo la recuperación realizada de manera sintáctica o semántica, usando principalmente fuentes de conocimiento léxicas u ontologías.

### **2.2.1.1. Recuperación de Servicios de Telecomunicaciones**

La recuperación de servicios de telecomunicaciones tiene en la actualidad algunas aproximaciones: redes de nueva generación en SOA para la recuperación y composición de servicios que buscan ofrecer capacidades de la red a terceras partes [71]. Otros enfoques basados en el protocolo SIP, que permiten la identificación y recuperación de servicios a partir de una solicitud SIP sobre arquitecturas IMS [72], [73].

Existen, por otro lado, proyectos orientados al usuario final que permiten la creación de servicios complejos y personalizados de acuerdo a requerimientos específicos de cada usuario. Los más importantes son los proyectos Europeos OPUCE (*Open Platform for User-centric service Creation and Execution*) [21] y SPICE (*Service Platform for Innovative Communication Environment*) [22]. Estos proyectos se encuentran orientados al usuario final debido al acompañamiento y soporte ofrecido mediante el uso de contexto y preferencias de usuario. Sin embargo, es de resaltar que la recuperación de servicios en entornos de telecomunicaciones, sigue siendo una tarea difícil para usuarios sin los conocimientos técnicos necesarios.

En contraste a la Web, la recuperación de servicios en el dominio Telco, trabaja notablemente sobre una cantidad mucho menor de servicios, ya que las capacidades de telecomunicaciones son, por lo general, definidas por los mismos desarrolladores del sistema de recuperación.

### **2.2.1.2. Recuperación de Servicios basada en Propiedades No Funcionales**

Un aspecto que no se ha considerado en los trabajos anteriores y en general en la mayoría de los trabajos de recuperación de servicios, es la recuperación con base en las propiedades no funcionales de los servicios. Aunque es claro el progreso en los últimos años, existe una cantidad reducida de trabajos que lo consideran [61]. Varios trabajos sobresalen por su estudio en esta área. El primero es [62], el cual gira en torno a un sistema que extiende el catálogo de negocios de internet UDDI (*Universal Description, Discovery and Integration*), agregando una base de datos en la arquitectura orientada a servicios, la cual puede ser consultada dando nombres de propiedades funcionales y no funcionales para obtener archivos WSDL (*Web Service Description Language*) de servicios que se ajustan a la consulta. Otro de los trabajos destacados alrededor de este tema es [63], el cual adapta el método LSP [64] (*Logic Scoring Preferences*), originalmente creado para tomar decisiones en problemas de múltiples criterios como la selección de hardware. La adaptación de este método le permite a los autores, a partir de pesos definidos por los usuarios, reflejar relaciones de dependencia entre preferencias no funcionales; es decir, determinar situaciones donde, con base en la selección de propiedades no funcionales, se requiere el cumplimiento de todos o la mayoría de criterios, o únicamente se requiere el cumplimiento de los requerimientos seleccionados.

Existe un conjunto representativo de enfoques que consideran la recuperación de servicios, teniendo en cuenta las propiedades no funcionales, en el dominio Web. Sin embargo, en la literatura son escasos los enfoques que consideran las propiedades no funcionales para la recuperación de otro tipo de recursos como los servicios de telecomunicaciones. Esta limitante resulta clave a la hora de recuperar servicios en entornos convergentes, ya que las capacidades de telecomunicaciones evidencian propiedades no funcionales mejores en términos generales, en contraste a servicios en la Web que implementen una misma funcionalidad (llamadas, mensajería instantánea, etc.). Por esta razón, el considerar preferencias no funcionales en el proceso de recuperación garantiza una selección de servicios más acorde a los requerimientos del usuario.

## **2.2.2. Procesamiento de Lenguaje Natural**

En esta sección se abordan los trabajos relacionados que procesan solicitudes escritas ó habladas, en lenguaje humano como el inglés, con el objetivo de usarlos en diversos sistemas como los de recuperación de servicios. Dentro de las propuestas realizadas, se identifican dos grupos de acuerdo al nivel de complejidad y aproximaciones utilizadas para el procesamiento de la solicitud: procesamiento débil y procesamiento fuerte.

### **2.2.2.1. Procesamiento Débil de Lenguaje Natural**

Los trabajos de esta categoría se caracterizan porque los mecanismos de PLN utilizados, se limitan a funcionalidades básicas de procesamiento, cuyos objetivos se orientan al análisis secundario de una solicitud: número de palabras, identificación de palabras mal escritas, categorías de las palabras, separación de frases, etc. Por tanto, se puede afirmar que estos mecanismos, aunque importantes para el soporte de sistemas de recuperación de servicios orientados al usuario final, no tienen un impacto fuerte como otras tareas más complejas de PLN. Algunos de los trabajos que se encuentran en esta categoría son [51], [59], [60]. En [51] se utiliza WordNet para medir la similitud semántica de dos frases. Una de las tareas usadas es la de Desambiguación de Sentido de Palabras (*Word Sense Disambiguation*) a través del algoritmo de Lesk, este algoritmo se fundamenta en la idea de que palabras que coocurren en una frase son usadas para referirse a un mismo tema, y que usualmente sentidos relacionados de palabras son definidos en un diccionario usando las mismas palabras [43]. Por otra parte, en el trabajo descrito en [57], las solicitudes realizadas en lenguaje natural son representadas como un conjunto de palabras clave debidamente etiquetadas, las cuales son usadas para un proceso de recuperación de servicios plenamente sintáctico. Para la implementación de este enfoque, los autores usaron diversas técnicas como lematización y WSD (Word Sense Disambiguation) [57]. Para el caso de [60], los autores presetan un analizador de solicitudes cuya finalidad es identificar nombres, verbos y adjetivos de solicitudes en lenguaje natural realizadas en texto plano.

### **2.2.2.2. Procesamiento Fuerte de Lenguaje Natural**

Las propuestas en este grupo se caracterizan por tener mecanismos de PLN que generan un impacto fuerte en el soporte de sistemas de recuperación de servicios orientados al usuario final. La funcionalidad de los sistemas aborda el PLN como componente funcional de la recuperación de servicios [4], [9], [48], [52], [54], [74]. En [45], se reemplaza una solicitud formal, hecha por un diseñador en un formato definido donde se describen clases de anotaciones semánticas como propiedades funcionales y no funcionales, por un agente inteligente encargado de analizar solicitudes realizadas en lenguaje natural (inglés) para extraer información semántica, específicamente las metas (Las metas son representaciones de un objetivo, cuyo cumplimiento se solicita a través de la ejecución de un servicio Web), y de ellas derivar información semántica adicional, como entradas y salidas, las cuales son usadas por el mismo agente para recuperar servicios e informar el orden de composición de los mismos.

El trabajo descrito en [4], propone un modelo para la composición automática de servicios, enfocándose principalmente en resolver el problema de identificación de servicios, mediante el análisis de la solicitud realizada en lenguaje natural. Dentro del análisis lingüístico de la solicitud del usuario, se pasa por procesos de segmentación de texto, remoción de palabras irrelevantes, lematización (método para reducir una palabra a su raíz) y corrección gramatical. Asimismo, implementa una medida de distancia conceptual, basadas en grafos de conceptos, para calcular la similitud entre la solicitud del usuario, representada por palabras clave y otra serie de palabras que describen a los servicios disponibles en un repositorio.

La supercomputadora de IBM, WATSON [48], realiza fases de análisis a preguntas realizadas en lenguaje natural a través de cientos de algoritmos. WATSON contesta preguntas que exigen un razonamiento profundo dado a su grado de complejidad, en menos de tres segundos. El proyecto usa técnicas de PLN, una de las tareas más interesantes es la realizada por un parser [49] que convierte el texto a una forma

más estructurada: un árbol que muestra una estructura superficial y una estructura lógica profunda. Dentro de esta estructura es posible observar, por ejemplo, la separación de texto indicando categorías gramaticales de las palabras. Otro de los mecanismos usados por WATSON, es la detección de reglas usando el lenguaje Prolog [50], con el fin de detectar términos claves de la pregunta, como las palabras que referencian la respuesta, tipo de entidad que se está preguntando (p.ej., poeta, político, obra de arte, etc.) y la relación correspondiente entre estas entidades.

En [52], se desarrolla un sistema para el reconocimiento de entidades (*Named Entity Recognition* - NER), el cual permite identificar nombres propios (entidades) en un texto y asociarlos con tipos comunes (p.ej. lugares, personas, nombres, organizaciones, etc.). Los autores de [52] combinan métodos de aprendizaje supervisados y no supervisados. Los métodos supervisados consisten en un sistema que lee largos textos anotados, memoriza listas de entidades y crea reglas basadas en características discriminativas [53]. Por otra parte, los métodos no supervisados no requieren largos textos anotados para su aprendizaje, ya que tienen definida su funcionalidad. Los métodos usados en [52] se basan en el modelo de máxima entropía (método de aprendizaje supervisado), redes neuronales (método de aprendizaje supervisado) y reglas de selección de patrones (método de aprendizaje no supervisado).

El trabajo desarrollado en [54], integra dos técnicas de aprendizaje automático (*machine learning*) para la definición de un sistema NER del lenguaje árabe. Las técnicas usadas son un clasificador de reconocimiento de patrones semi-supervisado y otro de campos aleatorios condicionales (*Conditional Random Fields*: CRF) supervisado. En resumen, CRF es un modelo probabilístico [54], usado para segmentar y etiquetar datos secuenciales; el objetivo es encontrar términos que maximicen cierta secuencia de términos. Para el desarrollo del sistema NER, se emplearon adicionalmente herramientas para el etiquetado gramatical de palabras y una base de datos léxica árabe.

Cuando se habla de recuperación de recursos a partir de lenguaje natural, Siri no puede dejar de ser mencionado. Siri es un asistente personal que puede encontrarse como aplicativo en algunos dispositivos iPhone y es capaz de realizar un conjunto de tareas, entendiendo instrucciones habladas en lenguaje natural. Basándose en información de contactos, librerías de música, calendario, recordatorios y centros de datos de Apple, Siri puede entender las solicitudes y devolver respuestas [74], las cuales están relacionadas con las aplicaciones incorporadas en el iPhone, incluyendo clima, mensajería, contenido multimedia, entre otros. Sin embargo, al ser una aplicación privativa, restringe el acceso a recursos permitidos por Apple, es decir, no cualquier servicio puede ser consumido desde el dispositivo. Adicionalmente, y de acuerdo a pruebas funcionales realizadas sobre la aplicación, se pudo observar que Siri no considera las propiedades no funcionales de los recursos que recomienda a los usuario [43].

### 2.2.3. Brechas Existentes

#### 2.2.3.1. Esquema de evaluación del Estado del Arte

Dada la cantidad de trabajos relacionados con la presente investigación, se ha definido un esquema de evaluación como el observado en la Tabla 2. De esta forma, se puede visualizar y analizar con mayor facilidad las brechas existentes y por ende, las ventajas de los enfoques propuestos alrededor de la recuperación de servicios orientada al usuario final. Para esto, se han definido factores de evaluación, los cuales evidencian la carencia o cumplimiento de propiedades en los trabajos descritos en la sección 2.2, estos son: **Soporte al Usuario (SU)**, es decir, si posee mecanismos que permitan ofrecer facilidades al usuario al momento de solicitar servicios y que hagan el sistema más intuitivo; **Automático (A)**, corresponde al nivel de intervención del usuario con el sistema para realizar todo el proceso; **Propiedades No Funcionales (PNF)**; **Propiedades Funcionales (PF)**; **Lenguaje Natural (LN)**; y finalmente **Escalabilidad a Entornos Convergentes (EEC)**, el cual indica la capacidad del sistema para recuperar otro tipo de recursos, independiente de la red de acceso del mismo.

Cada criterio fue evaluado en las propuestas según 4 niveles: *Alto (F)*, *Medio (M)*, *Débil (D)* y *Ausente (A)*; indicando el nivel de cumplimiento con respecto a un criterio determinado, tal como se muestra en la Tabla 2.

**Tabla 2. Comparación de Propuestas de Trabajos Relacionados.**

Enfoques	SU	A	PNF	PF	LN	EEC
1. Bosca et al. (2006)	Alto	Medio	Ausente	Medio	Medio	Alto
2. Pop et al. (2010)	Alto	Medio	Ausente	Medio	Medio	Alto
3. Sangers (2009)	Alto	Medio	Alto	Medio	Alto	Medio
4. Magableh (2011)	Medio	Medio	Ausente	Ausente	Alto	Medio
5. Sutthikulphanich (2008)	Alto	Medio	Medio	Medio	Medio	Medio
6. IBM (2011)	Medio	Medio	Ausente	Ausente	Medio	Ausente
7. Gawinecki et al (2010)	Medio	Medio	Medio	Medio	Ausente	Medio
8. Mallick et al (2010)	Alto	Medio	Ausente	Medio	Alto	Medio
9. Nawaz et al (2008)	Ausente	Medio	Ausente	Medio	Ausente	Alto
10. Dinsing, et al (2007)	Medio	Alto	Ausente	Ausente	Ausente	Medio
11. Carlin et al. (2010)	Ausente	Alto	Ausente	Medio	Ausente	Medio
12. Falcarin (2009)	Ausente	Medio	Alto	Medio	Ausente	Medio
13. Al-Masri et al. (2009)	Medio	Alto	Medio	Alto	Ausente	Alto
14. Qing et al. (2008)	Medio	Medio	Medio	Alto	Ausente	Alto
15. Schmidt et al. (2008)	Ausente	Medio	Ausente	Ausente	Medio	Ausente
16. Seon et al. (2001)	Ausente	Medio	Ausente	Ausente	Medio	Ausente
17. AbdelRahman et al.(2010)	Ausente	Medio	Alto	Alto	Medio	Ausente
18. Apple (2011)	Medio	Medio	Alto	Medio	Medio	Medio

SOPORTE

Alto
Medio
Bajo
Ausente

### 2.2.3.2. Discusión

A pesar de que los trabajos relacionados de la Tabla 2 corresponden a enfoques claramente diferenciados en la sección 2.2.1, se puede observar que la mayoría de los frameworks desarrollados, no ofrecen un soporte consistente al usuario. Para los trabajos centrados en la recuperación de servicios, esto significa que la mayoría de trabajos no tienen un componente alto de acompañamiento al usuario y se concentran más bien en el proceso de recuperación, restando importancia a la participación del usuario en la definición de sus requerimientos. Adicionalmente, los sistemas orientados al usuario final, principalmente los que cuentan con componentes de PLN, han conseguido soportar de alguna manera al usuario final, empleando algunos mecanismos. Sin embargo, estas aproximaciones no han logrado ofrecer un soporte consistente, ya sea por restringir las consultas realizadas en lenguaje natural, ofrecer mecanismos de soporte que no garantizan la mejor usabilidad del sistema ó ser sistemas cuya complejidad de uso sigue siendo elevado (P.ej. sistemas que aceptan solicitudes en lenguaje natural pero requieren una constante retroalimentación para recuperar los servicios ó sistemas cuyo resultado no es claramente entendible por los usuarios). Cuando se menciona soporte consistente se hace referencia a la simpleza y fácil manejo (abarcando los mecanismos necesarios) garantizados por el sistema, para que un usuario final pueda recuperar servicios por si solo considerando una mínima curva de aprendizaje.

En este sentido, es posible observar la necesidad de brindar un mayor acompañamiento al usuario final en este tipo de actividades. En cuanto al uso de propiedades funcionales y no funcionales para los trabajos de recuperación de servicios, es posible concluir que, en su mayoría, las propuestas se basan en preferencias funcionales, dejando de lado los requisitos no funcionales, los cuales pueden llegar a aportar gran sentido a las descripciones semánticas de los servicios. La decisión de no considerar parámetros no funcionales es problema común, sobre todo para los enfoques centrados en la recuperación de servicios Web, ya que los parámetros no funcionales no son considerados como un factor crítico, a diferencia del dominio de las telecomunicaciones, donde estas propiedades si son relevantes. Sin embargo, la importancia de los parámetros no funcionales en los procesos de recuperación ha crecido hasta ser considerados componentes necesarios. Pese a esto, los enfoques revisados que consideran parámetros no funcionales, muestran debilidades bien sea para brindar soporte al usuario ó para usar mecanismos de PLN.

En sistemas de recuperación de servicios orientados al usuario, las aproximaciones basadas en mecanismos de PLN que fueron revisadas, juegan un papel importante. La razón es que pueden afectar la usabilidad de un sistema de recuperación de servicios de manera crítica, al dejar por fuera el tratamiento de solicitudes de mediana complejidad; por ejemplo, solicitudes con condicionales y conectores, o de una mayor complejidad: como frases sin estructuras predefinidas, ya que se ignoran términos que pueden facilitar la recuperación de servicios con mayor precisión. De esta forma, a pesar de tener mecanismos de recuperación de servicios eficientes, la confiabilidad del sistema se ve comprometida por la fase inicial de captura de la solicitud. Conscientes de esto, es importante brindar mecanismos de PLN que no sean tan restrictivos a la hora de capturar la solicitud del usuario.

Debido a la tendencia de los usuarios a usar cualquier tipo de servicio (Web o Telecomunicaciones), es necesario soportar la escalabilidad a entornos convergentes. Sin embargo, pese a los aportes de los enfoques que han desarrollado frameworks para la recuperación de servicios, no todos y en especial aquellos que brindan un soporte medio/alto al usuario son escalables a estos entornos, surgiendo problemas para garantizar una recuperación de servicios en entornos convergentes, en especial a usuarios que no poseen conocimientos técnicos.

Aún con los esfuerzos realizados por parte de los trabajos estudiados, la recuperación de servicios orientada al usuario, presenta inconvenientes que deben ser resueltos, entre los que se encuentran: soporte al usuario más consistente, captura flexible de requerimientos, escalabilidad en la recuperación de recursos (aplicaciones, servicios, etc) y consideración de propiedades funcionales y no funcionales para el proceso de recuperación.

Por lo anterior, considerando las observaciones realizadas en esta sección, el presente proyecto propone un conjunto de técnicas y algoritmos para soportar la recuperación de servicios en entornos convergentes orientado al usuario final, y de esta forma, aportar en la solución de los diferentes inconvenientes detectados.

### **2.3. RESUMEN**

Este capítulo presentó el estado actual sobre investigaciones desarrolladas alrededor de la recuperación de servicios sobre entornos convergentes centrada en el usuario final, comenzando por contextualizar al lector, brindándole una descripción de conceptos generales, con el fin de comprender el campo de aplicación del presente proyecto. Posteriormente, fueron descritos los diferentes trabajos relacionados, organizados en enfoques generales con base en las diferentes temáticas que abarca el proyecto. Finalmente, un esquema de evaluación fue definido con el fin de identificar brechas y aportes de los diferentes trabajos, analizando los enfoques estudiados y posicionando la propuesta del presente trabajo de grado.

### **3. ANÁLISIS DE LENGUAJE NATURAL.**

En este capítulo se describen los elementos constitutivos de una importante rama de la Inteligencia Artificial (IA): el Procesamiento del Lenguaje Natural (PLN). El PLN está orientado a facilitar la comunicación hombre-computadora por medio del lenguaje natural (denominado también lenguaje humano) y puede concebirse como una disciplina encargada de reconocer y utilizar la información expresada en dicho lenguaje (a través de la voz o texto) dentro de los sistemas informáticos.

El PLN evidencia desde hace varios años un fuerte desarrollo debido a sus múltiples e importantes aplicaciones, como: recuperación de información, acceso a bases de datos, corrección automática, sugerencia de palabras, resumen automático, traducción automática, creación semiautomática de mapas conceptuales, detección de sentimientos, entre otras.

De tal forma, en este apartado, inicialmente, se realiza una descripción conceptual del PLN. Seguido, son definidos los componentes de un sistema de PLN, para posteriormente seleccionar y justificar aquellos que son necesarios en el desarrollo del presente proyecto. Por último, algunas de las tecnologías y plataformas existentes en la actualidad son descritas, así como también las respectivas adaptaciones y modificaciones que fueron realizadas con el fin de cumplir con los objetivos del presente trabajo de grado.

#### **3.1. PROCESAMIENTO DE LENGUAJE NATURAL**

La recuperación de información mediante el procesamiento de las solicitudes realizadas por un usuario final es una de las tareas clave abordadas en el presente trabajo de investigación. Este proceso consiste esencialmente en escoger un grupo de elementos que tengan relevancia en una consulta. Actualmente las técnicas más usadas para la recuperación de información, involucran la identificación de palabras clave dentro de un documento por medio de su frecuencia de aparición en el texto analizado. Dentro de este análisis, el modelo probabilístico y algunos procedimientos iterativos e interactivos son los más usados, estos buscan identificar elementos importantes en una solicitud y encontrar un sentido general a la petición, para poder comprender a mayor profundidad las necesidades de un usuario en particular.

En [75] se introduce un modelo que brinda una descripción detallada de la solicitud en lenguaje natural realizada por un usuario final (ver Figura 4). Esta contribución, fruto de la presente propuesta de investigación, expone las diferentes dimensiones que pueden ser distinguidas a partir de tal solicitud: palabras de control, propiedades funcionales y no funcionales. El elemento más extenso y relevante corresponde a las propiedades funcionales de la solicitud, que a su vez, y según la naturaleza de la misma, puede llegar a contener parámetros de comportamiento y de entrada/salida; de igual forma, esta propiedad se extiende para cada una de las sub-categorías de esta dimensión. Por otro lado, se encuentran las palabras de control y los parámetros no funcionales; todo este conjunto de elementos y su importancia en el proceso de recuperación de servicios, será explicado de manera detallada en secciones posteriores.

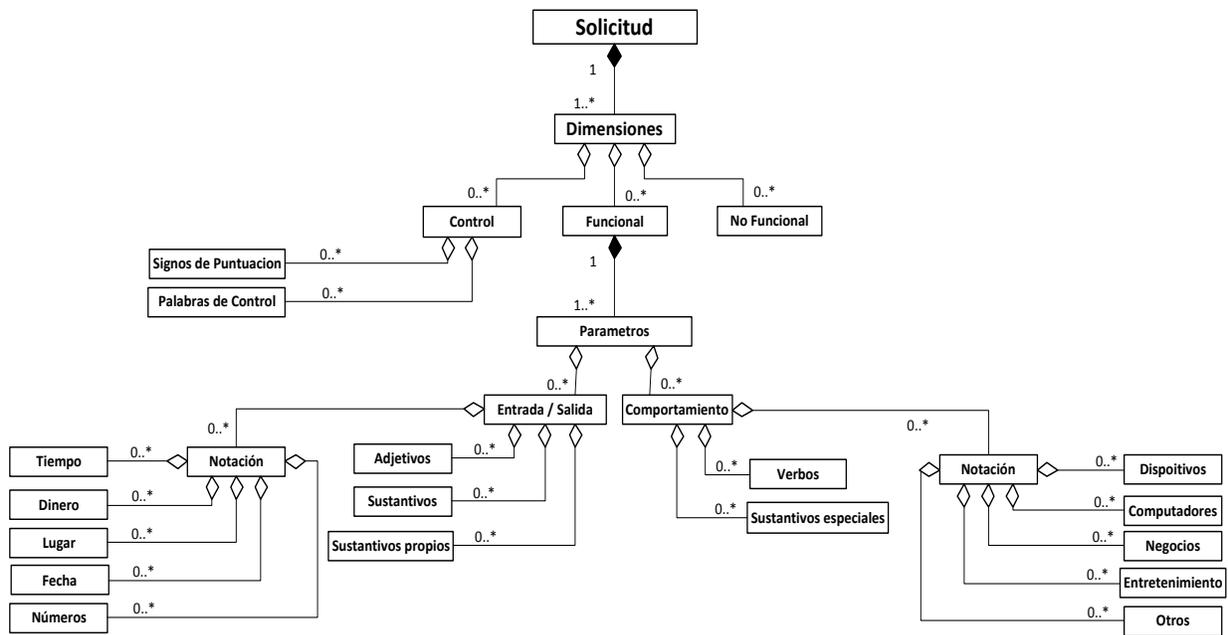


Figura 4. Modelo de una Solicitud en Lenguaje Natural

### 3.1.1. Modelo Conceptual

La Figura 5 expone el modelo conceptual de PLN propuesto, basado en un enfoque híbrido, el cual relaciona los conceptos generales estudiados de la sección 2. Los niveles de análisis lingüístico escogidos se muestran en la figura a continuación, éste modelo no contiene los niveles pragmático y discurso, ya que con los 5 niveles identificados es posible obtener los elementos relevantes de la solicitud.

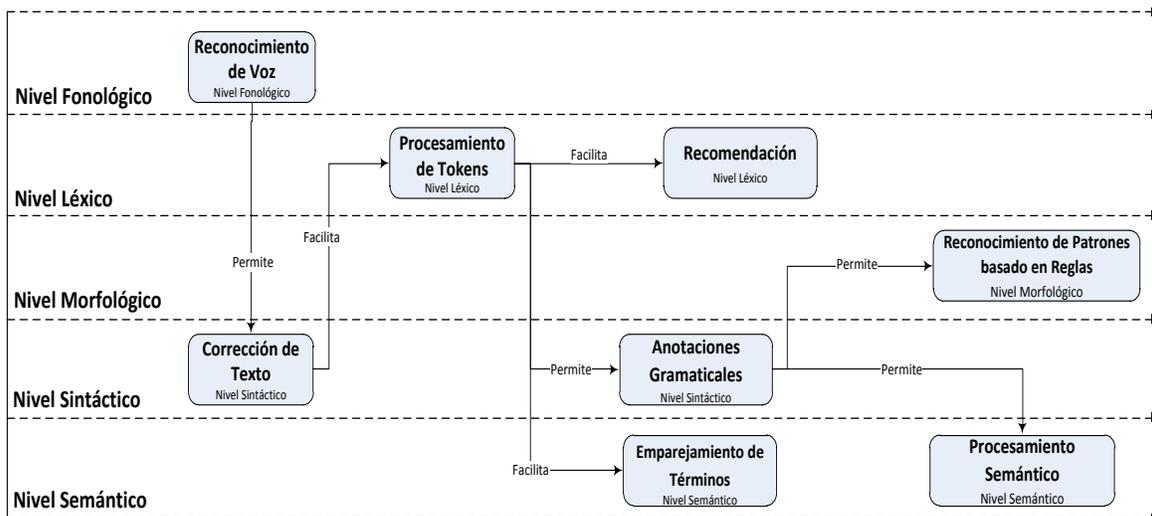


Figura 5. Modelo Conceptual

El usuario puede realizar solicitudes en voz y texto. En el caso de solicitudes por voz, el primer paso es *procesar la entrada a texto*; posteriormente se *corrigen* las palabras indicando al usuario si la solicitud posee errores de escritura o sin significado en el idioma elegido (inglés). Tras la corrección, se divide el texto en unidades básicas conocidas como *tokens*, con el fin de analizar cada término por separado y someterlo a un proceso de *anotación gramatical*. Este procedimiento resulta útil para realizar un

*procesamiento semántico* a través de la desambiguación de términos y ejecución de algoritmos de *reconocimiento de patrones basados en reglas*. Estos últimos tienen el propósito de identificar términos importantes para la recuperación de servicios ó encontrar variaciones de cada término y complementar las solicitudes. Así mismo, se realizan *recomendaciones de términos* que otros usuarios, con peticiones similares, han utilizado para recuperar servicios. Este proceso es útil porque permite la participación y soporte de la comunidad de usuarios cuando uno de ellos desea un servicio.

Otro de los procesos realizados a los tokens, es el *emparejamiento*, que consiste en comparar los términos recuperados de la solicitud, con listas que definen diferentes categorías, de esta forma, en caso de coincidir un término con algún ítem de las listas, el término es clasificado en la categoría y puede ser utilizado para definir la lógica de flujo de los servicios ó como componente importante de la solicitud del usuario para el proceso de recuperación de los servicios más adecuados.

Cada uno de los componentes descritos anteriormente, poseen mecanismos específicos de PLN que serán descritos en las siguientes secciones.

### **3.1.2. Niveles de análisis lingüístico**

El análisis lingüístico de una solicitud realizada en lenguaje natural, puede plantearse desde varios niveles lingüísticos, donde diversos estudios [81], [162] señalan una serie de pasos mínimos que se deben cumplir con el objetivo de llevar a cabo un análisis correcto. En primer lugar debe efectuarse un reconocimiento del idioma, posteriormente la segmentación del texto en palabras u oraciones facilita el manejo de elementos; seguido, debe realizarse un análisis morfológico, sintáctico y finalmente uno semántico. Para el caso de estudio, fue considerado el inglés como idioma objetivo, gracias a la facilidad y extensibilidad que presenta, por lo cual la identificación del idioma como primera tarea dentro del procesamiento de lenguaje natural, es omitida. A continuación cada uno de los niveles lingüísticos son definidos, detallando su importancia en el desarrollo del presente trabajo.

#### **3.1.2.1. Análisis Lingüístico a Nivel Fonológico.**

La fonología es una de las tareas más complejas en el procesamiento del lenguaje natural, la cual estudia las unidades fónicas o fonemas [82], [83] y su análisis puede conllevar, en algunos idiomas, a disertaciones verdaderamente complejas. Sin embargo, para el caso del inglés<sup>3</sup>, su nivel de dificultad no es de gran magnitud dada la estructura gramatical que posee, en consecuencia, el estudio de la parte material de los sonidos y el análisis de las propiedades físicas de los mismos se torna altamente factible.

El aparato de fonación a través del habla, es por esencia un generador de señales, objeto de estudio en la fonología, hecho que ha desencadenado en la generación de tecnologías de reconocimiento de voz, que permiten obtener un texto a partir de la voz humana. Habitualmente, el reconocimiento de voz es utilizado como una interfaz entre el humano y la computadora por algún tipo de software y debe cumplir con las siguientes tareas [161]:

- *Procesamiento*: el cual convierte la señal de voz a una forma que el reconocedor pueda procesar (Conversión señal análoga a digital).
- *Reconocimiento*: Identifica lo que el humano trata de decir (traducción de señal digital a texto).
- *Comunicación*: Envía lo reconocido al sistema de software de aplicación.

Adicionalmente son muchos los desarrollos tecnológicos a nivel de software reconocedor de voz. Existe una amplia gama en diferentes categorías como:

---

<sup>3</sup> Disponible en: <http://www.usingenglish.com/weblog/archives/000037.html>

- *Dispositivos móviles*: Sonic Cloud, Dragon Dictation and Search, Google Voice Search, Bing voice, GoVivace, Siri Personal Assistant, MeMeMe Mobile Speech Recognition, Shoutout, DriveSafe.ly, Vlingo, Jeannie, Speaktoit and Ziri Assistant Beta for Android, Microsoft Tellme and Ask Ziggy for Windows Phone 7.
- *Open source*: CMU Sphinx, Julius, simoni, ATROS, RWTH ASR, SHoUT.
- *Dispositivos Macintosh*: Dragon dictate, Speakable items, ViaVoice, Voice Navigator.
- *Dispositivos Windows*: VoiceAttack, Voice Finger, WSR Toolkit, TrigamTech, Vocola, Windows Speech Recognition.

Dado a que el número de tecnologías es amplio y su funcionalidad es bastante óptima, en la presente propuesta se hizo uso de la tecnología Google Voice Search, la cual se implementa en un dispositivo móvil Android, con el fin último de poder ampliar el rango de usuarios que puedan hacer uso de la aplicación e ingresar las solicitudes de una manera más ágil y cómoda.

### **3.1.2.2. Análisis Lingüístico a Nivel Léxico.**

En la mayoría de las ocasiones este análisis es conocido como el primero dentro de un sistema de procesamiento de lenguaje natural y obtiene como salida componentes independientes, que en muchos casos son unidades léxicas y/o símbolos [84].

En este campo de estudio, existen diversas tareas de suma importancia que atacan diferentes componentes a través de la subdivisión automática de un texto en unidades más cortas. Por ejemplo, el análisis de segmentación en temas, está relacionado con la obtención de un conjunto de palabras relacionadas entre sí y que tratan de una misma idea en particular. Asimismo, se encuentra el análisis de segmentación de una oración, el cual mediante la identificación de signos de puntuación, obtiene el conjunto de palabras que componen cada oración seleccionada. Dada la naturaleza del presente trabajo, los anteriores tipos de análisis no son relevantes, debido a que no son necesarios para ninguna de las etapas del procesamiento de análisis de lenguaje natural que se llevan a cabo.

Dentro del Nivel Léxico existe otra tarea denominada *Tokenización*. Este proceso es de suma importancia, ya que tiene como fin dividir una cadena de texto, para obtener elementos más pequeños y así trabajar con ellos de una manera más óptima. Estas unidades atómicas se denominan *tokens*, las cuales representan palabras, símbolos o números dentro de un texto. Frecuentemente la *tokenización* se produce a nivel de palabras, sin embargo, muchas veces es difícil identificar qué es en realidad una palabra, para lo cual los analizadores usan métodos iterativos de búsqueda más avanzados, que hacen uso de reglas con enfoque heurístico, con el propósito de identificar los tokens deseados.

La importancia del proceso de *tokenización*, recae sobre su uso en otras tareas de PLN, como el etiquetado semántico, volviéndolo en ocasiones una función indispensable en los sistemas de PLN. Dentro de las implementaciones de esta tarea, existe una gran gama de tecnologías útiles para realizar este proceso de *tokenización*, entre las que se encuentran: el tokenizador de Stanford (Stanford Tokenizer [85]), el tokenizador incluido en GATE e incluso debido a la facilidad de implementación existen decenas de códigos en línea que permiten realizar esta función [86].

### **Recomendación lingüística a nivel léxico.**

Tras la *tokenización*, se obtienen palabras útiles para un proceso de recomendación. Cada vez que un usuario ingresa las primeras letras de una palabra y haciendo uso de una técnica de autocompletado, es posible sugerir diversas palabras que coinciden con el *token* que está siendo ingresado. Esta operación de autocompletado, en sistemas de recuperación de servicios, trae como beneficio que los usuarios ingresen peticiones con palabras que estén directamente relacionadas con los servicios, lo que conlleva a una generación de un ranking más preciso, ligado siempre a los requerimientos expresados por del usuario.

Adicionalmente, con los *tokens* obtenidos, es posible obtener otro tipo de recomendación relacionada con servicios o flujos, siempre que estos estén relacionados con términos utilizados previamente por otros usuarios, con el fin de llevar a cabo un proceso de emparejamiento sintáctico simple.

### **3.1.2.3. Análisis Lingüístico a nivel Morfológico.**

Las palabras son conocidas en el mundo lingüístico, como una unidad y es la morfología, la encargada de estudiar la estructura interna de dichas unidades, con el fin de identificar, analizar y describir la estructura de las mismas.

Puede suceder que las palabras ingresadas por los usuarios en sistemas de recuperación de servicios, use diversos vocablos complejos, que en muchas ocasiones resultan difíciles de procesar. Por tal motivo, es necesaria la implementación de técnicas que permitan obtener palabras más simples que faciliten su procesamiento. Es aquí donde la tarea de *lematización* (en inglés Stemming) cobra importancia. Esta tarea obtiene unidades lingüísticas básicas con significado semántico (lema) a partir de un término flexionado (es decir palabras conjugadas, plurales o con género).

Un *lema*, es formalmente aceptado como representante de todas las formas flexionadas de una misma palabra; es decir, el lema de una palabra es la que usualmente se encontraría en un diccionario tradicional (p.ej., singular para sustantivos, masculino singular para adjetivos, infinitivo para verbos) [87]. Por ejemplo, “*decir*” es el lema para palabras como: *dije, diré, dijéramos, dijo*.

El proceso de *lematización* puede llevarse a cabo de diversas maneras, sin embargo la mayoría de técnicas o *lematizadores* automáticos que existen en la actualidad, hacen uso de alguno de estos cuatro enfoques para llevar a cabo dicho proceso:

1. *Búsqueda en una tabla*: hace uso del denominado árbol-B o dispersión de búsqueda; consiste en comparar las palabras originales con una tabla donde se almacenan todos los términos índice y sus lexemas respectivos.
2. *Eliminación de Afijos*: elimina los sufijos y/o prefijos de los términos.
3. *Lematizadores de variedad de sucesores*: usan, como base para la lematización, las frecuencias de las secuencias de letras en un cuerpo del texto.
4. *Métodos de n-gramas*: unen los términos con base en el número de bigramas o n-gramas que comparten.

La implementación de un proceso como el de *lematización* cuenta con diferentes ventajas y debilidades. Entre las ventajas principales se encuentra:

1. *Eficiencia*. Se ve directamente reflejado en los procesos de recuperación de información.
2. *Comprensión*. Para tareas de procesamiento de lenguaje natural, no se pierde información en la mayoría de las veces, sobre los conceptos a tratar cuando se trabaja con un proceso de lematización.
3. *Mejora las consultas*. Los procesos de consulta se pueden acercar mucho más a lo esperado.

Aunque el porcentaje de éxito en los procesos de *lematización* es alto, existen algunas desventajas que deben ser consideradas cuando se haga uso de esta tarea, ya que puede distorsionar el significado de una petición cuando el análisis requerido sea altamente estricto.

Este tipo de desventaja puede ser considerada especialmente cuando se desencadenan errores al momento de obtener el *lema* de una palabra y son los denominados a continuación:

- *Hiperlematización* (en inglés Over-Stemming): exceso en la eliminación de caracteres en una palabra generando lemas carentes de sentido.

- *Hipolematización* (en inglés Under-Stemming): insuficiencia en la eliminación de caracteres, generando lemas muy parecidos a la palabra original.

Para el proceso de *lematización* existe un conjunto de algoritmos como *Lovins stemmer*, *Porter stemmer* y *Paice stemmer*, así como también otros métodos basados en análisis que hacen uso de diccionarios (análisis lexicográfico) como por ejemplo: *KSTEM*, *stemming con cuerpo*, entre otros.

El uso de algoritmos es la forma más adecuada, debido a la exactitud y facilidad en el manejo de los procesos. Dentro de los algoritmos, uno de los más reconocidos para el proceso de lematización, de acuerdo a parámetros de desempeño, documentación y reconocimiento es el algoritmo de Porter [88], [89]. Este algoritmo consiste esencialmente en un proceso de remoción de sufijos morfológicos comunes e inflexionados, aplicado principalmente en palabras en inglés, pero que perfectamente puede ser extendido a otros idiomas. Este algoritmo fue desarrollado por el laboratorio de computación de Cambridge en 1979 [91].

En cuanto a las tecnologías que han realizado un desarrollo de software en esta materia, se encuentra uno de los más representativos, llamado Snowball [90], el cual es un lenguaje que soporta la representación de diversos algoritmos de lematización como el algoritmo de Porter.

Adicionalmente, entre las múltiples ventajas de usar Snowball, se encuentra que es posible generar código en ANSI C, Java y cuenta con un respaldo amplio tanto en idiomas (12 en total) como en documentación electrónica. Estos motivos son suficientes para tenerlo como herramienta principal en el proceso de lematización, cuyo objetivo adicional en sistemas de recuperación de servicios es brindar mejores resultados, al enriquecer la solicitud del usuario con las palabras en su forma no flexionada.

#### **3.1.2.4. Análisis Lingüístico a nivel Sintáctico.**

El análisis sintáctico en el campo de la Lingüística, es el análisis de las funciones sintácticas o relaciones de concordancia y jerarquía que guardan las palabras agrupadas entre sí, en oraciones. La importancia de realizar este tipo de análisis, radica principalmente en la interpretación y comprensión de los textos, especialmente de documentos problemáticos de ámbitos como la legislación, la política o la tecnología.

La tarea de PLN que sobresale en este nivel, es el etiquetado gramatical o en inglés POST (Part of Speech Tagging). El POST puede ser entendido como el proceso que asigna o etiqueta palabras de un texto de acuerdo a su respectiva categoría gramatical (artículo, sustantivo, pronombre, verbo, adjetivo, adverbio, preposición, conjunción, interjección), lo cual puede realizarse basándose en la definición de la palabra sobre el contexto en el que aparece o por la relación con las palabras adyacentes en una oración [92].

Para el presente trabajo es importante realizar el proceso de etiquetado, ya que con él se puede identificar el sentido correcto de una palabra que presente ambigüedad en el contexto de la solicitud presentada por un usuario. Además, establecer las diferentes categorías gramaticales de un conjunto de palabras en una petición, ayuda a comprender y analizar mejor la estructura de las solicitudes de un usuario.

Para llevar a cabo esta tarea de etiquetado sintáctico, existe una amplia gama de desarrollos, algunas de las tecnologías más populares son: Brill Tagger, CLAWS POS tagger, Junk Tagger, MedPost, TnT POS Tagger, Transformation-Based Tagger, TreeTagger.

En el presente trabajo, es utilizado el etiquetador sintáctico ofrecido por GATE. Este etiquetador está basado en el etiquetador Brill Tagger; sin embargo, GATE ha realizado una serie de adaptaciones, con el objetivo de optimizar sus condiciones y ofrecer un etiquetador que se ajuste mas a la verdadera categorización gramatical de las palabras.

### **3.1.2.5. Análisis Lingüístico a nivel Semántico.**

El estudio semántico, hace referencia a aspectos de significado, sentido e interpretación de un determinado concepto [94]; en términos generales puede ser entendido como el estudio de la codificación del significado dentro de una expresión lingüística [93].

La polisemia y la homonimia son dos conceptos y/o fenómenos presentes dentro del análisis semántico. La polisemia puede ser entendida como el fenómeno lingüístico, en el cual un mismo ítem léxico tiene más de un significado asociado. En la polisemia se puede presentar lo que se conoce como ambigüedad, representada por una expresión que tiene más de una interpretación y es el contexto el que soporta habitualmente la selección de uno de los significados posibles. Por otro lado, La homonimia representa la coincidencia en la forma de dos términos, pero su etimología y su significado son totalmente diferentes [95].

En muchas ocasiones las palabras pueden referirse a diferentes elementos dependiendo del contexto en el que se esté manejando. Por ejemplo, la palabra en inglés “bank”, puede referirse a una entidad financiera o al objeto para sentarse. Es aquí donde surge un problema de dualidad en el significado, para lo cual es necesario establecer un proceso de desambiguación semántica que permita identificar cuál es la definición acertada. En consecuencia, dentro del análisis semántico se hace necesaria la implementación de la tarea de desambiguación semántica, encargada de determinar el sentido correcto de una palabra. Esta tarea puede involucrar una serie de procesos, para lo cual *Ide y Véronis* en [103] han definido dos operaciones dentro del proceso de desambiguación:

1. Determinación de los diferentes sentidos de una palabra.
2. Asignación a cada ocurrencia de una palabra su sentido adecuado.

Es decir, en una primera fase se desarrolla un “perfil” de los sentidos de cada palabra a partir de los contextos asociados a la misma y posteriormente, se identifica el perfil que se le ajusta más en cada contexto.

Estas fases y en general el proceso de desambiguación semántica, brindan para el presente trabajo aportes como: poder realizar una expansión de los términos usados por los usuarios en la solicitud de un servicio, a través de los términos relacionados en la definición del sentido correcto de una palabra, lo cual es de suma relevancia para mejorar la exactitud en procesos de emparejamiento.

Con el fin de seleccionar el mecanismo de desambiguación más adecuado, fueron identificadas metodologías que se basan en inteligencia artificial ó en fuentes de conocimiento estructuradas y no estructuradas. Para el último caso, una de las herramientas más conocidas es Prolog; este lenguaje de programación está diseñado para representar y utilizar el conocimiento que se tiene sobre un determinado dominio. Aunque su implementación ha sido alta, presenta algunas limitaciones que nuevas tecnologías han tratado de solucionar. Tales limitaciones están relacionadas con problemas ligados a la representación del conocimiento, incapacidad ocasional de reconocer que un problema es inaplicable, limitantes en los motores de inferencia, entre otras [96].

Adicionalmente, existen varios algoritmos para llevar a cabo estos enfoques, que van desde algoritmos genéticos hasta algoritmos simples. Estas aproximaciones pueden identificar el significado correcto de una palabra a partir de la categoría gramatical asociada y del uso de un diccionario léxico. El algoritmo más conocido perteneciente a este último es el algoritmo de Lesk [97].

El algoritmo de Lesk, además de ser uno de los más antiguos, es también uno de los algoritmos mas implementados gracias a su confiabilidad, facilidad y simplicidad de implementación (compacto). El algoritmo de Lesk es comúnmente entendido como el procedimiento que determina los sentidos de las palabras que ocurren en un contexto particular, basándose en una medida de ambigüedad que se encuentra presente.

Formalmente, para encontrar la relación semántica entre dos conceptos de acuerdo a lo definido por el algoritmo de Lesk en [98], se debe considerar el mayor solapamiento que viene dado por el número de palabras en común que resultan a partir de las definiciones de dos palabras. A continuación se muestra la descripción del proceso que hace el algoritmo de Lesk, a través de las siguientes ecuaciones:

Dadas dos palabras  $\{W_1, W_2\}$  y  $S_1$  y  $S_2$  sus definiciones respectivamente, se tiene: el primer paso en la ecuación 1, muestra como a partir de las definiciones  $S_1$  y  $S_2$ , es posible obtener el conjunto de palabras relevantes para realizar el siguiente proceso de análisis:

$S_1$  y  $S_2$  son asumidos como los sentidos correctos para:

$$Puntaje_{Lesk}(S_1, S_2) = gloss(S_1) \cap gloss(S_2) \quad (1)$$

Donde la función *gloss*, obtiene el conjunto de palabras que contiene la definición  $S_i$  del concepto  $W_i$ .

Adicionalmente, el cálculo de similitud de las definiciones de los sentidos evaluados, se maximizan de acuerdo al valor que toma el solapamiento de glosas mostrado en la ecuación 2 a continuación:

$$Solapamiento\ de\ glosas = |Sentido\{W_1\} \cap Sentido\{W_2\}|. \quad (2)$$

Ahora bien, si el proceso mostrado anteriormente requiere extensión a un contexto, donde no solamente contenga un número pequeño de palabras sino de  $n$ , es necesario calcular y realizar la productoria señalada en la ecuación 3:

$$\prod_{i=1}^n Sentido\{W_i\} \quad (3)$$

Una variante, denominada Lesk Mejorado, fue propuesta en [99]. Esta nueva propuesta considera no solo las definiciones de las palabras a desambiguar, sino también las definiciones de aquellos términos relacionados semánticamente en la jerarquía WordNet. Como esta variante existen muchas más; sin embargo para el presente trabajo se ha implementado una adaptación del algoritmo de Lesk señalado en [104], con el objetivo de obtener una definición más precisa de una palabra ambigua y con ello enriquecer futuras etapas de emparejamiento semántico presentados en este trabajo.

Por otro lado, como complemento de esta tarea de análisis lingüístico, se incluye un proceso importante perteneciente igualmente al nivel lingüístico semántico. Este proceso surge a partir de la necesidad de identificar palabras de la solicitud del usuario dentro de alguna categoría con el fin de detectar términos relevantes para el proceso de recuperación de servicios. Este proceso abarca la clasificación de términos en determinadas entidades.

La tarea de clasificación de términos en determinadas entidades consiste principalmente en la detección de conceptos dentro de listas que incluyen nombres (Juan, María, Pedro), organizaciones (ONU, OEA, IBM), expresiones de tiempo (horas, minutos, segundos), unidades monetarias (\$100, pesos, dólares), lugares (Europa, Bogotá, parque) entre otros tipos; sin embargo, estas entidades pueden ampliarse dependiendo del campo de acción con el que se esté tratando. Es así, como estas categorías de palabras cobran relevancia, permitiendo que, a partir de una petición, se pueda determinar y encontrar términos relacionados con algún tipo de clasificación previamente definido y con ello simplificar y mejorar el tratamiento del lenguaje.

En este orden de ideas, formalmente se ha definido esta clasificación como el Reconocimiento de Nombres de Entidad o en inglés: N.E.R. (por sus siglas *Name Entity Recognition*). Dentro de la propuesta del presente trabajo, se definieron tres categorías superiores (ver Figura 4: Funcional, No Funcional y Control) y dos subcategorías importantes en la clasificación funcional (ver Figura 4: Entrada/Salida y

Comportamiento) con el fin de tener mayor eficiencia en los procesos de emparejamiento entre los términos introducidos por los usuarios y las etiquetas que definen los servicios Web y de telecomunicaciones.

Como lo muestra la Figura 4, se estableció la categoría de comportamiento (*Behavior*), la cual se encarga de identificar aquellos elementos que representen verbos o características propias de la funcionalidad de un servicio Web o de telecomunicaciones. Por otra parte, se tiene la categoría de Entrada/Salida (*Input/Output*), donde se concentran todas aquellas palabras relacionadas con las interfaces (entradas y salidas) de un servicio. Estas categorías junto con las demás posibles clasificaciones existentes (fecha, locación, tiempo, entre otras), identifican de acuerdo al análisis semántico de una oración, su categoría.

Dentro de las categorías definidas es posible identificar criterios no funcionales que se encuentran inmersos en la solicitud, esto se da porque las propiedades no funcionales pueden ser entendidas y expresadas como adjetivos por parte de algún usuario inexperto. De este modo el sistema identifica este tipo de propiedades no funcionales genéricas dentro de la solicitud de un usuario.

Para la clasificación de términos en entidades, existen varios enfoques que están relacionados con los basados en ingeniería de conocimiento y aprendizaje automático [100], este último debido a su complejidad de implementación no es tenido en cuenta.

El enfoque basado en ingeniería de conocimiento, es el más utilizado en este tipo de procesos debido a los buenos resultados obtenidos. En su implementación, existen dos formas comunes que son: uso de expresiones regulares y los conocidos Gazetters. Las expresiones regulares tratan con la sintaxis de las frases que constituyen los nombres de entidad. Los Gazetters usualmente son conocidos por contener un conjunto de listas, en las cuales se encuentran diversos contenidos que hacen referencia a una colección de elementos agrupados (instituciones, colores, nombres, ciudades, etc.) [101].

Gazetter y JAPE (Java Annotation Patterns Engine) [102] son los elementos usados en este trabajo. JAPE es un componente Open Source, también conocido como un transductor de estado finito, el cual aborda la generación de reglas gramaticales con el fin de obtener anotaciones sobre diversas palabras agrupadas en las listas Gazetter. La tecnología N.E.R. mencionada, ha sido empleada para seleccionar y clasificar el conjunto de palabras de una petición de entrada, en dos categorías de importancia que representan las Entradas/Salidas y atributos de Comportamiento de servicios Web y de Telecomunicaciones.

## **3.2. PLATAFORMAS PARA EL PROCESAMIENTO DE LENGUAJE NATURAL**

Para realizar un procesamiento de los elementos expuestos anteriormente, se han generado algunas plataformas donde son expuestas varias tecnologías, para llevar a cabo un estudio sobre las diferentes tareas de lenguaje natural. A continuación se muestra una lista con algunas de las más reconocidas plataformas, dada su amplia documentación, soporte y facilidades para trabajar en diferentes tareas de análisis de lenguaje natural.

### **3.2.1. Apache UIMA (*Unstructured Information Management Applications*)**

UIMA<sup>4</sup> (Aplicaciones para la administración de información no estructurada) es una herramienta de código abierto (Open Source), desarrollada por Apache<sup>5</sup>, cuyo objetivo es apoyar todos los avances en cuanto a contenido no estructurado como: texto, audio y video. Sus herramientas pueden ser implementadas tanto en lenguaje JAVA como en C/C++, Perl y Phyton, ya sea en sistemas Linux o Windows, a través de un *plugin* de eclipse el cual, posterior al análisis desarrollado, obtiene un documento anotado de acuerdo a las características y tareas de lenguaje natural aplicadas.

---

<sup>4</sup> Disponible en: <http://uima.apache.org/>

<sup>5</sup> Disponible en: <http://www.apache.org/>

Entre los reconocimientos más importantes que ha tenido, se encuentra el sistema de inteligencia artificial denominado Watson [48], un supercomputador desarrollado por IBM, que participo en un show televisivo norteamericano, que lo convirtió en un sistema famoso capaz de responder preguntas muy complejas.

Dentro de las capacidades más importantes que ofrece esta tecnología se encuentra:

- Tokenizador anotado.
- Etiquetador que hace uso de Snowball.
- Etiquetador de expresiones Regulares.
- Etiquetador que hace uso del modelo oculto de Markov
- Etiquetador BSF
- Etiquetador de diccionario.
- Etiquetador OpenCalais.
- Indexador Lucene CAS
- Concepto anotador de Mapper.
- Extractor de Características – configurable.

### 3.2.2. OpenNLP.

OpenNLP<sup>6</sup>, es otra herramienta provista por Apache la cual soporta la mayoría de las tareas de procesamiento del lenguaje natural, como: tokenización, segmentación de oraciones, etiquetado sintáctico o POS tagging, reconocimiento de nombres de entidad NER y resolución de correferencia. Todas estas tareas son requeridas para el análisis avanzado de sistemas de procesamiento de lenguaje natural.

Entre las ventajas que ofrece OpenNLP se encuentra la capacidad de ser integradas con otro tipo de software de PLN. El requerimiento de procesamiento y recursos es bajo en comparación con otras tecnologías y es adoptado por desarrolladores e investigadores que trabajan bajo Open-source.

### 3.2.3. GATE (*General Architecture for Text Engineering*)

GATE<sup>7</sup> corresponde a una suite de herramientas para el procesamiento de lenguaje natural desarrolladas por la universidad de Sheffield, las cuales se utilizan en la actualidad por una gran comunidad de desarrolladores, estudiantes, empresas y científicos. Este reconocimiento se debe al gran soporte brindado para aplicaciones orientadas al PLN. Algunas de sus funcionalidades son:

- Modelado y persistencia de estructuras de datos especializadas.
- Medidas, evaluación, benchmarking.
- Visualización y editado de anotaciones, ontologías, árboles de análisis sintáctico, etc.
- Un idioma de transducción de estados finitos.
- Extracción de instancias de entrenamiento para aprendizaje automático.
- Implementaciones de aprendizaje automático.

Su nivel de aplicación está fuertemente relacionado con la bio-informática y sistemas de recuperación de información. Entre las principales características que presenta GATE se destaca su sistema de extracción de información denominado ANNIE (A Nearly New Information Extraction Sysytem). ANNIE ha sido ampliamente implementado tanto a nivel empresarial (sistemas industriales) como a nivel académico (proyectos mundiales conocidos como: MUC, TREC, ACE, DUC, Pascal, NTCIR). Además contiene herramientas orientadas a la extracción de información e incorpora un amplio abanico de recursos para los distintos niveles de análisis del lenguaje.

---

<sup>6</sup> Disponible en: <http://openlp.org/>

<sup>7</sup> Disponible en: <http://gate.ac.uk/>

Dentro de GATE, los *plugins* constituyen los componentes de desarrollo de los sistemas de PLN y corresponden a Java Beans conocidos como CREOLE (Collection of Reusable Objects for Language Engineering). Estos objetos pueden ser de tres tipos:

1. *Recursos lingüísticos*: son elementos no algorítmicos que sirven como contenedores de información (diccionarios, esquemas de anotación, corpora y documentos).
2. *Recursos de procesamiento*: estos elementos son unidades de procesamiento de datos (POS taggers, stemmers, tokenizadores, separadores de frases, reconocedores de entidades, etc).
3. *Aplicaciones*: son componentes que permiten integrar recursos de procesamiento para realizar operaciones diversas sobre textos. De esta forma se puede construir programas que toman un corpus y obtienen el POS de los documentos que contiene.

Existen unos modos de trabajo que GATE incorpora en su funcionamiento:

1. Un modo gráfico que carga una interfaz programada en Java, y desde la cual se pueden hacer varias operaciones de PNL.
2. Mediante una API para Java, permite utilizar clases y librerías con funcionalidades de tareas de PLN.

### 3.2.4. NLTK (Natural Language Toolkit)

Es un conjunto o una suite de librerías escritas y desarrolladas en Python. NLTK<sup>8</sup> incluye dentro de su documentación un amplio desarrollo gráfico y datos de ejemplo, que a su vez son acompañados de una extensa documentación sobre desarrollos y API de soporte. La tecnología incluye un libro que detalla los conceptos más relevantes detrás del procesamiento de lenguaje natural y asimismo ofrece una serie de cursos a través de los cuales las personas pueden adquirir más conocimiento en la materia.

Para desarrollos que sean de carácter educativo, NLTK se convierte en la herramienta ideal para aquellas personas que están apenas introduciéndose en el aprendizaje de PLN o estén comenzando a investigar en él. En términos generales, NLTK puede considerarse una herramienta o plataforma de enseñanza para las tareas más importantes del PLN; sin embargo, no ha sido tomada en cuenta para construir prototipos de gran envergadura.

### 3.2.5. Discusión.

Como puede observarse son diversas las tecnologías que se han desarrollado para llevar a cabo un proceso más completo en el análisis de lenguaje natural. Sin embargo, muchos de los enfoques, aunque presentan diversas propiedades y ventajas, contienen una serie de limitaciones bien sea por su baja capacidad a nivel de análisis del lenguaje, o por su bajo nivel de documentación y desarrollo, aspectos relevantes para la consecución del presente trabajo.

En este sentido, es imprescindible conocer los puntos negativos y positivos de este conjunto de herramientas, y de manera global, compararlas de acuerdo a características de desempeño y factibilidad, según lo expuesto en [105].

**Nota:** algunas propiedades se califican de 1-3, donde 3 es la máxima ponderación de la propiedad, adicionalmente (X) representa la existencia de la propiedad para una determinada tecnología, mientras que (-) significa la carencia de la misma.

---

<sup>8</sup> Disponible en: <http://nltk.org/>

Tabla 3. Comparación Herramientas para el procesamiento de Lenguaje Natural

Herramientas	Ubicación		Lenguaje de programación				Modelo de licencia	Documentación			
	Paquete de descarga	Software libre	Java	Python	C/C++	Otro	Modelo de licencia	Manual de instalación	Tutoriales	Libros	Ejemplos
Rapidminer [152]	X	X	X	-	-	-	AGPL	1	2	-	-
NLTK [153]	X	X	-	X	-	-	GPLv2	2	2	X	2
Gate [154]	X	X	X	-	-	-	LGPL	2	2	-	1
OpenNLP [155]	X	X	X	-	-	-	Apache v2	3	3	-	-
WEKA [156]	X	X	X	-	-	-	GPL	2	2	X	3
JWebPro [157]	X	X	X	-	-	-	GPL	3	3	-	-
Apache UIMA [158]	X	X	X	-	X	X	Apache	1	2	-	1
LingPipe [159]	X	X	X	-	-	-	RFL	2	2	-	-
Orange [160]	X	X	-	X	X	-	GPL	2	1	-	-

Tabla 4. Comparación Herramientas para el procesamiento de Lenguaje Natural II

Herramientas	Windows	Linux/Unix	MacOS X	Configuració	Usabilidad	GUI	Visualizació	Comunidad Activa
Rapidminer [152]	X	X	X	2	2	X	2	2
NLTK [153]	X	X	X	2	2	-	-	2
Gate [154]	X	X	X	2	2	2	3	2
OpenNLP [155]	X	X	X	1	1	-	-	1
WEKA [156]	X	X	X	2	2	2	1	2
JWebPro [157]	X	X	X	1	1	1	-	2
Apache UIMA [158]	X	X	X	2	1	1	1	2
LingPipe [159]	X	X	X	2	1	-	1	1
Orange [160]	X	X	X	3	2	-	-	1

Como se puede destacar de las Tablas 3 y 4, GATE es una de las herramientas que presenta mejores valores en cuanto las características que se evalúan. Entre los atributos más representativos se destacan los elementos de: configuración, usabilidad, comunidad de desarrollo, lenguaje de programación, modelo de licencia y el sistema multiplataforma que ofrece. Estos elementos junto con las propiedades que se especificaron en cada una de las plataformas descritas, se suman a los criterios establecidos para la selección de GATE como plataforma principal de desarrollo del presente trabajo de grado. En el Capítulo 5 se ampliará la descripción de la plataforma de PLN escogida, detallando sus características más relevantes.

### 3.3. EJEMPLO

Para facilitar la comprensión de las tareas de análisis de lenguaje natural, se cita en esta sección un ejemplo que detalla los procesos realizados en esta fase. Para ello, se considera inicialmente la siguiente

situación: un ejecutivo entendido como un usuario inexperto en procesos de recuperación de servicios, hace uso del lenguaje natural para expresar, desde su dispositivo móvil, la siguiente solicitud:

*“Deseo recibir veinte minutos antes de la reunión, el reporte del tráfico de Bogotá a través de mensajes, después de la reunión quiero recibir el contenido de audio de las decisiones tomadas”.*

Una vez el usuario introduce la petición, ésta llega y es analizada inicialmente por la primera tarea de análisis de lenguaje natural, la cual corresponde a la tokenización como se ha definido en la Sección 3.1.2.2. El resultado de esta tarea, tiene como salida “tokens” identificados de forma individual, los cuales por razones didácticas, se muestran a continuación separados por un guion (-), con el propósito de ilustrar el procedimiento.

*“Deseo-recibir-veinte-minutos-antes-de-la-reunión-reporte-del-tráfico-de-Bogotá-a-través-de-mensajes...”*

Seguidamente, dentro del análisis léxico, el sistema hace un filtro de palabras, con lo cual se pretende remover aquellos términos que no aportan significado al proceso de análisis de lenguaje natural (por ejemplo: *yo, la, quiero, deseo*, etc.) o que pueden incrementar el tiempo de procesamiento en operaciones futuras siendo innecesarias. El resultado omitiría las palabras que coincidan con un conjunto de palabras previamente identificadas y tendría como resultado una lista de “tokens” mas reducido como el siguiente:

*“Recibir-veinte-minutos-antes-reunión-reporte-tráfico-Bogotá-mensajes-despues-reunión...”*

Una vez obtenido el conjunto de palabras relevantes, se realiza un proceso de etiquetado gramatical. Esta tarea, la definida en la Sección 3.1.2.4. como POST, es una implementación del etiquetador de GATE, la cual además, adiciona al *token* una etiqueta con su categoría gramatical respectiva, tal como se señala a continuación:

*“recibir: Verbo (VB) – veinte: Adjetivo (ADJ) - minutos: Sustantivo (NN) – antes: Adverbio (ADV) – reunión: Sustantivo (NN)...”*

En la Sección 3.1.2.5, se definió una tarea capaz de clasificar palabras de acuerdo a un conjunto de reglas predefinidas denominada NER (Reconocimiento de Nombres de Entidad). Esta tarea, a partir de las etiquetas gramaticales definidas anteriormente y las listas de entidades propias del sistema, identifica (asignando una etiqueta) cuáles términos pueden hacer parte de las categorías: *Funcional* (Comportamiento, Entrada/salida) y *Control*. Adicionalmente, a partir de la detección de una palabra de control, se identifican bloques dentro de la solicitud, los cuales agrupan algunos términos, tal como se muestra en el siguiente ejemplo:

*“Bloque1: **Recibir:** funcional-comportamiento, **veinte:** funcional-entrada/salida...  
**después:** Control  
Bloque 2: **reunión:** funcional-comportamiento...”*

Adicionalmente, en el evento en que un término de la solicitud presente uno o varios significados, el sistema a través del algoritmo modificado de desambiguación semántica de Lesk, identifica cuál de las definiciones de una palabra es la correcta. Por ejemplo, la palabra tráfico, posee varios significados, ya que puede ser una conjugación del verbo traficar, o puede referirse al sustantivo que hace alusión a la agregación de las cosas (peatones, vehículos, narcóticos, armas, etc.) que van y vienen de un punto a otro durante un período específico de tiempo. Ahora bien, como se pudo ver anteriormente, la palabra “tráfico”, posee más de un significado consistente, por lo cual se procede a evaluar cada uno de sus significados para obtener finalmente el significado correcto dentro de la solicitud. Sí el significado correcto de la palabra “tráfico” identificado fuese: “flujo de vehículos en una vía, calle o autopista...”. Entonces el sistema selecciona uno de los primeros sustantivos de la definición como “vehículo”, con el objetivo de enriquecer semánticamente

el significado de la palabra “tráfico”, la cual va a ser tomada en cuenta en el proceso de recuperación de servicios.

Posteriormente se procede a pasar las palabras por una tarea de *lematización* expuesta en la Sección 3.1.2.3. Aquí, a través del uso del proyecto Snowball, se identifican los *lemas* o palabras raíces, como por ejemplo:

“mensajes: **mensaje**, decisiones: **decision**, minutos: **minuto**...”

Finalmente, el proceso de análisis de lenguaje natural obtiene como salida un conjunto de palabras relevantes identificadas con una serie de etiquetas que muestran a qué categoría pertenece o qué identificador gramatical tiene asociado; estos elementos se utilizan en la siguiente etapa de recuperación de servicios señalada en el ejemplo del capítulo 4.

### 3.4. RESUMEN

En esta sección se presentaron los conceptos y enfoques específicos del PLN, abordando la descripción del modelo propuesto para representar una solicitud en lenguaje natural, con el fin de mostrar la clasificación de términos y los tipos a los que pertenecería cada término de una solicitud, tras aplicar las diferentes tareas de PLN seleccionadas.

Posteriormente, se realiza una recopilación de los niveles de análisis lingüísticos existentes. Luego de un análisis exhaustivo de cada uno de ellos, fueron seleccionados los niveles: sintáctico, léxico, fonológico, semántico y morfológico, dada su importancia y el ofrecimiento de mecanismos que facilitan el proceso de análisis de solicitudes realizadas en lenguaje natural. Consecuentemente, se desarrollaron diversas tareas dentro de cada uno de los niveles expuestos. Tales tareas corresponden a: *tokenización*, *autocompletado*, *lematización*, etiquetado de parte del discurso POS, reconocimiento de nombres de entidad *NER* y desambiguación semántica *WSD*. Del mismo modo, se presentaron un conjunto de plataformas disponibles, en las cuales se podría desarrollar cada una de las tareas descritas para cada nivel.

Tras el análisis de las diferentes propiedades (configuración, usabilidad, comunidad de desarrollo, lenguaje de programación, modelo de licencia y sistema multiplataforma) de las plataformas, la selección de las tareas y la identificación de los requerimientos, fue posible elegir las tecnologías y mecanismos más adecuados en cada caso, teniendo como plataforma base a GATE.

Finalmente, la sección presenta un ejemplo donde se ilustran las tareas seleccionadas y su respectiva funcionalidad, a través del procesamiento de una petición en lenguaje natural, realizada por un usuario desde su dispositivo móvil. El ejemplo muestra la entrada y salida de cada una de las tareas de análisis de lenguaje natural y describe asimismo el proceso (modificaciones, etiquetado, etc.) que van sufriendo los elementos constitutivos de la solicitud.

## 4. RECUPERACIÓN DE SERVICIOS ORIENTADO AL USUARIO FINAL

Este capítulo gira en torno al proceso de recuperación de servicios considerando propiedades funcionales y no funcionales. En primer lugar, se describen los problemas más relevantes relacionados con el vocabulario usado en las solicitudes y en las descripciones de servicios que afectan el proceso de recuperación de servicios; consecuentemente, se abordan los enfoques propuestos que buscan resolver los problemas descritos. Posteriormente, se presenta el estudio y selección de los enfoques más adecuados para generar un ranking de servicios con base en el procesamiento de la solicitud del usuario. Adicionalmente, se describe cómo el ranking mencionado es refinado con base en las propiedades no funcionales, obteniendo servicios más precisos, relevantes a la búsqueda del usuario. Finalmente, se realiza una descripción detallada de la salida final del proceso de recuperación.

Es preciso señalar que el proceso de recuperación de servicios descrito en este capítulo, corresponde a un aporte realizado en el presente trabajo de grado y se basa en adaptaciones de otros trabajos, así como también en algunas definiciones propias.

Con el fin de dar una visión general del proceso de recuperación de servicios y sus componentes más importantes, la Figura 6 representa el modelo conceptual del proceso de recuperación propuesto.

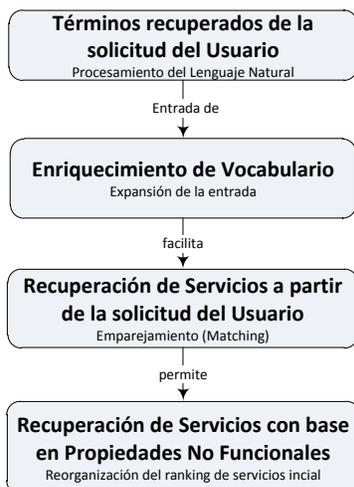


Figura 6. Flujo conceptual del proceso de recuperación de servicios

Los términos resultantes, tras el procesamiento de la solicitud con los diferentes mecanismos de PLN, pasan por un proceso de enriquecimiento para expandir los términos usando diferentes fuentes de información (WordNet<sup>9</sup> y Flickr<sup>10</sup>). Posteriormente, el conjunto de términos expandido es sometido a un proceso de emparejamiento (conocido en inglés como *Matching* [6]) con las descripciones de los servicios disponibles (publicados en un banco de recursos), generando así un ranking inicial de servicios, lo cual permite reducir el espacio de búsqueda del sistema. Finalmente, los servicios de este primer ranking son reacomodados teniendo en cuenta los valores de sus propiedades no funcionales y las preferencias no funcionales de los usuarios, obteniendo así el ranking final de servicios que responde a la solicitud del usuario final. Todo el proceso descrito, así como cada componente que lo soporta será visto en profundidad en el transcurso de este capítulo.

<sup>9</sup> Disponible en <http://wordnetweb.princeton.edu/perl/webwn>

<sup>10</sup> Disponible en <http://www.flickr.com/>

## **4.1. SOLUCIÓN A LIMITANTES DE VOCABULARIO**

La recuperación de servicios orientado al usuario presenta nuevas oportunidades y retos, para los enfoques tradicionales basados en la Arquitectura Orientada a Servicios (SOA), uno de los retos destacados es evitar que los proveedores de servicio tengan que proporcionar toda la información necesaria de los mismos [43], [106], con el fin de que los usuarios puedan recuperarlos.

En este contexto, cuando usuarios sin habilidades técnicas desean consumir un servicio, necesitan saber las palabras que lo describen, es decir, sus salidas, entradas y operaciones, o también su nombre exacto [43]; en caso de que el usuario proporcione descripciones erradas, se tiene una alta probabilidad de obtener servicios no deseados.

Para ello, se debe considerar el uso de un vocabulario más flexible, que permita contrarrestar limitantes como: ambigüedad de significados, diferente percepción de la funcionalidad del servicio por parte de los usuarios, sinónimo de términos, descripción insuficiente de servicios, entre otros; comprometiendo la funcionalidad del sistema con resultados de baja relevancia [107]. Es por esto que muchos de las propuestas actuales de recuperación y emparejamiento de servicios han tenido problemas, principalmente aquellos basados en la definición de servicios a través de archivos descriptores, como el Lenguaje de Descripción de Servicios Web (WSDL), debido a la falta de información sobre la funcionalidad de los servicios. En este contexto, actualmente existen algunos enfoques que permiten superar las restricciones de vocabulario mencionadas. A continuación, se presenta una breve descripción de las soluciones actuales más representativas.

### **4.1.1. Extracción semántica del vocabulario del usuario**

El proceso de extracción semántica del vocabulario, consiste en mirar las consultas de un usuario como si fueran estructuras semánticas emergentes, es decir, considerar que las consultas representan conocimiento de la persona y este conocimiento se encuentra relacionado de alguna manera, dependiendo de la interpretación dada acerca de lo que desea. Bajo esta consideración es posible usar las consultas de los usuarios para crear relaciones semánticas entre ellas y de esta manera generar ontologías livianas (especificación de conceptos representado por nodos, cuyas uniones –aristas- indican que tan relacionados semánticamente se encuentran [108]). De esta manera, las ontologías serían el resultado del sistema usado por los usuarios y no como tradicionalmente sucede (las ontologías son creadas por un grupo de expertos bajo un acuerdo común, consenso unificado de desarrollo), este concepto ha servido como base a una gran cantidad de trabajos para mejorar los procesos de recuperación [110], [111], [112], y [113].

Si bien los motores de inferencia son útiles para manipular términos y conceptos de acuerdo a reglas predefinidas, solo los usuarios que tienen la capacidad asociativa e interpretativa para crear y mantener representaciones de conocimiento como las ontologías [109], ya que requieren un actor confiable que prediga la manera cómo se interpretan los términos dentro de una comunidad de usuarios. Las ontologías derivadas del vocabulario utilizado por usuarios, pueden articular el conocimiento social en una forma procesable por las máquinas. Esto representa una notable mejora para la recuperación de información en los motores de búsqueda [114].

Lo expuesto permite superar los limitantes de vocabulario al describir un servicio a través de información útil suministrada por los usuarios, donde dicha información se encuentra estructurada mediante una representación formal de conocimiento (ontología) [65]. Sin embargo, este proceso, tanto a nivel académico como industrial, ha demostrado ser complejo, ya que, entre muchas otras restricciones, requiere de un gran esfuerzo en términos de desarrollo. Este problema se agrava aún más, puesto que la mayoría de las aproximaciones de este tipo evidencian fuertes restricciones a la hora de estructurar el vocabulario social en una ontología; p.ej., los términos usados deben estar organizados en clústeres para relacionar los significados entre clústeres y no entre los términos individuales [109].

Por consiguiente, las soluciones actuales expuestas atentan contra un criterio muy importante en los sistemas de recuperación de información orientada al usuario final y es el no interferir con los términos del usuario [115]. Algunas redes semánticas existentes [109] que emplean este proceso, ignoran la relevancia de conceptos individuales desde la perspectiva del usuario, lo que resulta en una imagen imprecisa del vocabulario usado por la comunidad. Finalmente, la mayoría de los sistemas bajo este enfoque, basan su funcionalidad en razonadores, mejorando la precisión pero intensificando a su vez el uso de recursos [116].

Debido a la naturaleza de las restricciones (limitantes) expuestas, fue necesario también considerar el mecanismo expuesto a continuación.

#### **4.1.2. Enriquecimiento semántico al vocabulario del usuario**

Este proceso normalmente utiliza fuentes de conocimiento para enriquecer semánticamente los términos usados por el usuario. La idea principal es expandir la búsqueda a partir de las palabras ingresadas en una solicitud, para relacionarlas con otras palabras a través de fuentes de conocimiento como ontologías ó sistemas de etiquetado. El proceso de enriquecer semánticamente el vocabulario del usuario resuelve problemas de ambigüedad y otras restricciones impuestas por el uso de vocabulario en lenguaje natural [115], es común encontrar diversos sistemas que siguen este enfoque [116], [117], [118], [119] y [120].

En general la consulta a vocabularios controlados ó fuentes sociales, permite extraer relaciones importantes que agregan significado a los términos usados por la comunidad de usuarios (P.ej. el término *wow* puede ser relacionado por la comunidad como un estado de asombro, otro ejemplo sería el significado a abreviaturas, cuando un usuario expresa en su estado un *lol* y la consulta permitiría indicar que se encuentra de buen humor). En contraste, fuentes de conocimiento, como ontologías, pueden solucionar la falta de relaciones semánticas entre los términos usados por el usuario y los términos que describen a un servicio (p.ej. sinónimos), además de proveer cierta organización al usar relaciones de especialización entre términos. Por otro lado, la falta de un contexto para definir términos propios del usuario, puede ser solucionado con sistemas provenientes de la Web Social, los cuales permiten definir un contexto al agregar etiquetas que desde la perspectiva de la comunidad se encuentran relacionadas y proveen un significado concreto, reduciendo la ambigüedad (P.ej. retomando, el término *wow* puede entenderse como un juego de video "World of Warcraft", grupo de mujeres escritoras "Women on Writing" ó un estado de ánimo. Los términos asociados en la web social corresponden a palabras como *wonderful* y *amazing* los cuales están relacionados con una expresión de asombro más que un juego ó un grupo, facilitando así la desambiguación del término).

La falta de relaciones semánticas en la consulta de servicios, es uno de los retos más grandes en los sistemas de búsqueda [115]. Por ello es necesario el uso de ontologías, ya que permiten obtener un conjunto de términos relevantes, que mejoran la relación semántica de la consulta del usuario, cuando solicita un servicio. Sin embargo, surgen ciertos problemas cuando la solicitud del servicio contiene términos que no son reconocidos por la ontología [115], p.ej. un servicio de mensaje de texto que el usuario describa como "SMS". Por lo tanto, es necesario considerar otro tipo de fuente de información que permita manejar este inconveniente. Los términos que no se encuentran en fuentes de conocimiento controlado (ontología), normalmente existen en sistemas de tipo social como: redes sociales (Twitter, Friends Reunited, Friendster, etc), sistemas de etiquetado (Flickr, delicious, etc). Es por esta razón que es conveniente usar las aplicaciones de medios sociales (*Social Media*) para extraer información que brinden mayor sentido y maximicen el significado de este tipo de términos.

En síntesis, por medio del uso de las fuentes de conocimiento descritas, es posible manejar los siguientes retos de vocabulario señalados en [115]: sinónimos de palabras, palabras polisémicas, formas alternativas de escribir un término, errores de escritura, términos demasiado específicos, escritura abreviada y relaciones semánticas.

#### 4.1.2.1. Selección de Ontología

Una de las de las posibles formas de clasificar ontologías es según el tipo de conocimiento contenido en ella [121]. Existen cuatro categorías en las que se puede clasificar las ontologías según este criterio:

- **De Representación:** son ontologías que definen los elementos básicos que conforman una ontología más específica. Uno de los ejemplos más representativos de esta clase de ontologías es el Frame-Ontology<sup>11</sup>.
- **Genéricas:** son meta-ontologías que definen términos generales que pueden ser usados en diferentes dominios. Un ejemplo interesante es la ontología de fútbol llamada FootbOWL<sup>12</sup>.
- **De Dominio:** describen conceptos relevantes de un dominio particular. BioTop<sup>13</sup> es una ontología de dominio utilizada en el dominio de la biología.
- **De Aplicación:** contienen los conceptos específicos necesarios en una aplicación. Una implementación muy popular es la ontología FMA<sup>14</sup> (Foundational Model of Anatomy).

Según la clasificación anterior, el interés del proyecto se centra en Ontologías Genéricas, ya que al estar orientado a usuarios con conocimientos técnicos limitados, es necesaria una fuente de conocimiento que no posea términos de un dominio específico. De esta forma, no se restringe la flexibilidad en el uso del lenguaje, proveyendo un gran número de conceptos que pueden contribuir a enriquecer los términos usados en la consulta de los usuarios.

Actualmente existen varias herramientas que pueden ser tratadas como ontologías de tipo genéricas, algunos ejemplos son CYC<sup>15</sup>, WordNet y Sensus<sup>16</sup>. La selección de una de estas herramientas está basada en factores como: accesibilidad, calidad y funcionalidad. Tras el análisis de estas herramientas, se decidió usar WordNet; las razones para su elección son expuestas a continuación: En primer lugar, su uso es gratis para propósitos de investigación [122]. Es una de las fuentes de conocimiento más populares para establecer relaciones entre términos [123]; además por su accesibilidad y calidad se ha convertido en la herramienta ideal para muchas aplicaciones como etiquetado semántico, recuperación de información entre otras [124]. Finalmente, a diferencia de varios lexicones existentes, WordNet tiene la ventaja de ser entendido tanto por humanos como por máquinas [115].

WordNet, además de ser un proyecto cuyo fin es comportarse como diccionario y tesoro para ser usado intuitivamente por usuarios finales, es también una ontología léxica [130], la cual representa una red de palabras conectadas entre sí según relaciones semánticas y léxicas. Las relaciones de tipo semántico se establecen entre significados de palabras, mientras que las relaciones de tipo léxico entre las formas de palabras. WordNet contiene sustantivos, verbos, adjetivos y adverbios, organizados dentro de conjuntos de sinónimos llamados Synsets, que a su vez, son usados para establecer conexiones entre ellos. Las principales relaciones semánticas son:

- **Relación Sinónimos:** dos palabras son sinónimos si la sustitución de una por la otra, en un contexto lingüístico no altera el significado [125].
- **Relación Antónimos:** el antónimo de una palabra X es la palabra Y perteneciente a la misma categoría gramatical que tiene un significado opuesto [125].
- **Relación Hipónimos:** es una relación de especialización entre una palabra específica y una más general (P.ej. un árbol es un tipo de planta) [125].

---

<sup>11</sup> Disponible en <http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/frame-ontology/index.html>

<sup>12</sup> Disponible en <http://www.taln.upf.edu/content/biblio/150>

<sup>13</sup> Disponible en <http://www.imbi.uni-freiburg.de/ontology/biotop/>

<sup>14</sup> Disponible en <http://sig.biostr.washington.edu/projects/fm/>

<sup>15</sup> Disponible en <http://www.cyc.com/>

<sup>16</sup> Disponible en <http://www.ilc.cnr.it/EAGLES96/rep2/node23.html>

- **Relación Merónimos:** es una relación compleja que se refiere a “*una parte de*” dado un sustantivo, p.ej. dedo es merónimo de mano [125].
- **Relación de Vinculación (*Entailment*):** es una relación unilateral de dependencia, p.ej. roncar está vinculada a dormir [125].
- **Relación Tropónimo:** permite relacionar verbos, parecida a los hipónimos de los sustantivos. P.ej. caminar es un topónimo de moverse [125].
- **Relación “*Similar a*”:** esta relación se concibió debido a que algunos adjetivos son similares en significado pero no lo suficiente para ponerlos dentro de un mismo *synset*, además no tienen significados opuestos para considerarlos antónimos, por lo que la solución fue relacionar esos adjetivos no antónimos y similares usando este tipo de relación [125].
- **Relación “*También ver*”:** esta relación que no está bien definida en la literatura, también es conocida como *términos relacionados*. En [125], los autores piensan que la realización de esta relación incluyó juicio humano para tomar caso por caso. Un ejemplo es: hostil con agresivo, violento y ofensivo.

Los problemas mencionados al inicio de la sección 4.1, son tratados con el uso de WordNet aplicado como una ontología. Las relaciones definidas en WordNet que se consideraron para el proyecto, corresponden a las mismas empleadas en [115], usadas en ese caso para mejorar semánticamente la búsqueda de recursos en sistemas de etiquetados orientados al usuario final. Las relaciones consideradas son: *Sinónimos*, *Hiperónimos-Hipónimos*, *Similar a*, y *También ver*.

#### 4.1.2.2. Selección de Fuente Social

Los términos que no se encuentran en Wordnet, suelen encontrarse en la Web Social [115]. Por ejemplo, WordNet no reconoce términos como “XML”, sin embargo, en las comunidades sociales tiene un significado. La idea con la selección de una fuente social es permitir agregar relaciones (P.ej. dar un contexto robusto a un término) a los términos usados por el usuario, particularmente aquellos no encontrados en una ontología. Una de las formas más comunes y simples de establecer relaciones semánticas en este tipo de comunidades, empleada también en este proyecto, es la agrupación (*clustering*) de términos. En sitios de la Web Social como Youtube, los términos utilizados para describir un video pueden ser agrupados en subconjuntos de términos llamados clústeres. Los grupos de términos presentan relaciones que fueron establecidas a partir de la interacción de los usuarios con el sistema. De esta manera, la creación de este tipo de clústeres, sirve como complemento a otras fuentes de conocimiento (p.ej., WordNet), permitiendo mitigar los problemas relacionadas con las restricciones de vocabulario, por ejemplo: palabras escritas por el usuario de manera abreviada o coloquial.

Para la implementación de un sistema de etiquetado Social que sigue el modelo de clústeres, primero se analizaron sistemas de etiquetado existentes en la Web que ofrecían esta característica en lugar de efectuar el clustering desde cero. De esta manera, de los sistemas estudiados, se seleccionaron los más populares e importantes en la actualidad: Delicious<sup>17</sup>, Flickr<sup>18</sup> y Youtube<sup>19</sup>. De esta manera, se establecieron unas características clave para escoger la solución más adecuada: una fuerte realimentación en el uso de etiquetas (es decir, la posibilidad que tiene la comunidad para influir en el uso de una etiqueta determinada), altos niveles de consenso por parte de los usuarios en el uso de las etiquetas respecto a un recurso y una variación media de vocabulario en el tiempo para definir las etiquetas.

En este sentido, según la investigación realizada en [126] sobre las características dinámicas de vocabulario en etiquetado social, los usuarios tienden a crear más etiquetas en Delicious que en los otros dos sistemas (Youtube y Flickr) durante un período de tiempo fijo. Seguido a este se encuentra Youtube y finalmente Flickr. Este comportamiento se debe al tipo de contenido que maneja cada uno, así como a la

<sup>17</sup> Disponible en <http://delicious.com/>

<sup>18</sup> Disponible en <http://www.flickr.com/>

<sup>19</sup> Disponible en <http://www.youtube.com>

tasa de crecimiento de recursos y usuarios, ya que para el caso de Delicious los usuarios pueden usar un vocabulario más variado por tratarse de páginas Web y no de videos (Youtube) o fotos (Flickr). Youtube por su parte, posee 10 veces más usuarios que Flickr, lo que se ve reflejado en las actividades de etiquetado de los videos. En contraste, Flickr al poseer una tasa más baja de crecimiento de etiquetas, presenta la posibilidad de obtener una realimentación colectiva más fuerte y un alto nivel de consenso sobre el tiempo [126].

Del trabajo desarrollado en [126] es posible concluir que: tanto Flickr como Delicious evidencian una mayor cantidad de usuarios en procesos colectivos de etiquetado e intercambio (*sharing*) de recursos, entre otros, a través de las funciones sociales que ellos proveen; mientras que en Youtube los videos tienden a ser etiquetados solo por los usuarios que los suben, limitando la variación de vocabulario. Flickr por su parte, no limita la variación de vocabulario como Youtube, pero tampoco permite un etiquetado libre, ya que restringe el proceso de etiquetar los recursos a únicamente usuarios con los permisos necesarios, haciendo que las etiquetas sean creadas por una cantidad menor de usuarios en relación a Delicious.

Según las características estudiadas, la fuente de servicios de medios sociales que más se ajusta a las características requeridas de este trabajo de grado es Flickr, razón por la cual se utilizaron los clústeres de este sistema, los cuales agrupan un conjunto de etiquetas relacionadas, de tal manera que utilizando un término popular es posible obtener los relacionados dentro de los clústeres, los cuales no tienen nombres y solo muestran las etiquetas que lo contienen. Esto hace posible al usuario explorar el espacio de información disponible de manera simple. Adicionalmente Flickr permite a los usuarios organizar el contenido de manera jerárquica dentro de taxonomías personales, agrupando los recursos (fotos) dentro de conjuntos (*sets*) y relacionándolos dentro de colecciones (*collections*). Así, la idea es enriquecer los términos usados por el usuario con términos encontrados en los clústeres que provee Flickr, con el fin de extender la descripción de los servicios desde la perspectiva de una comunidad.

Con base en el trabajo adelantado en [115], correspondiente al enriquecimiento de sistemas de etiquetado mediante fuentes sociales, se decidió usar el primer clúster (clúster: grupo de etiquetas relacionadas por la comunidad de usuarios) con las primeras tres etiquetas. Es decir, de los  $n$  clústeres obtenidos que relacionan una palabra, solo se consideró el primero, el cual contenía  $m$  etiquetas, de las cuales a partir de la cuarta, no existía una relación directa con el término del usuario, razón por la cual fueron ignoradas. Lo mismo ocurre con las etiquetas que se encontraron desde el segundo clúster [115].

De esta forma, con el fin de mejorar el proceso de recuperación de servicios fueron definidos dos procedimientos de enriquecimiento de vocabulario: Enriquecimiento de la descripción de los servicios y Enriquecimiento de la solicitud del usuario.

#### **4.1.2.3. Enriquecimiento de la Descripción de los Servicios**

Este procedimiento permite expandir los términos que describen los diferentes servicios (los cuales fueron obtenidos a partir de bancos de información de servicios web, especificaciones técnicas y anotaciones manuales, ver sección 4.2.1.2) con las relaciones semánticas de WordNet y Flickr en tiempo de compilación (antes de ejecutar el sistema por el usuario - *Offline*), con el fin de mejorar la información que describe a cada servicio e incrementar la precisión a la hora de realizar la búsqueda por parte de los usuarios.

#### **4.1.2.4. Enriquecimiento de la Solicitud de Entrada**

En tiempo de ejecución (*Online*), una vez se han capturado los términos relevantes que describen un servicio a partir de la solicitud realizada por el usuario en lenguaje natural, estos son sometidos a un proceso de expansión buscando relaciones semánticas usando WordNet y Flickr, con el objetivo de mejorar los resultados de búsqueda de los servicios.

## **4.2. RECUPERACIÓN DE SERVICIOS A PARTIR DE LA SOLICITUD DEL USUARIO**

Con la convergencia de los dominios Web y de telecomunicaciones alrededor del protocolo IP [43] y las nuevas tendencias en entornos de aplicación donde los usuarios son importantes generadores de contenido y aplicaciones, se abre un nuevo paradigma en el cual individuos sin capacidades técnicas tienen la habilidad de diseñar y crear sus propios servicios personalizados, integrando componentes del mundo Web y de telecomunicaciones. Como parte clave de este novedoso paradigma se encuentra la fase de recuperación de servicios, para la cual se han propuesto diversas alternativas con el fin de tornarla cada vez más eficiente. En la Sección 2.2.1, correspondiente al Estado del Arte, se detallan los trabajos más relevantes para la presente investigación. Con base en estos trabajos, a continuación se expone el sistema de recuperación de servicios propuesto.

### **4.2.1. Sistema de Recuperación de Servicios**

Con base en [118], para el presente proyecto se han adoptado características propias de los sistemas de recuperación de información orientados al usuario final, las cuales son expuestas a continuación:

- Facilidad de uso: simplicidad tanto para la creación de contenido como para su búsqueda.
- Instructivo: permite guiar la búsqueda.
- Interactivo: brinda la sensación al usuario de control y manejo de la aplicación.
- Adaptable: capacidad de adaptarse fácilmente a las preferencias del usuario.
- Presentación amigable: uso de componentes para una visualización comprensible (intuitiva) de contenidos y funcionalidades.
- Búsqueda exhaustiva: búsqueda de información completa considerando información de los usuarios como hábitos, forma de expresarse y áreas de interés.
- Precisión: capacidad para retornar resultados confiables que cumplan con lo solicitado por el usuario.

Tras analizar las principales representaciones de conocimiento descritas en la sección 2.1.4, se decidió que las folcsonomías por ser información creada por y para los usuarios (clasificación social), permiten soportar las características mencionadas anteriormente. Adicional a esto, facilitan la creación rápida y sencilla de contenido flexible y dinámico y al ser un sistema colaborativo, brinda mayor soporte al usuario. De esta forma, el mecanismo para describir los servicios y realizar la búsqueda de los mismos está basado en folcsonomías. La Figura 7, presenta el sistema general de recuperación de servicios, el cual relaciona los conceptos que van a ser presentados en el resto del capítulo para una mayor comprensión. El módulo de captura de requerimientos obtiene los términos que serán utilizados como entrada por el módulo de recuperación de servicios para encontrar los servicios más relevantes, para esto se tienen en cuenta las descripciones de los servicios realizadas con folcsonomías y los valores de las propiedades no funcionales de los servicios, generando un ranking final que asociado a términos de control obtenidos en la captura de requerimientos, crean un flujo de servicios, el cual es desplegado al usuario y almacenado para futuras recomendaciones. Adicionalmente el primer servicio de cada ranking generado es consumido y la respuesta obtenida es retornada al usuario.

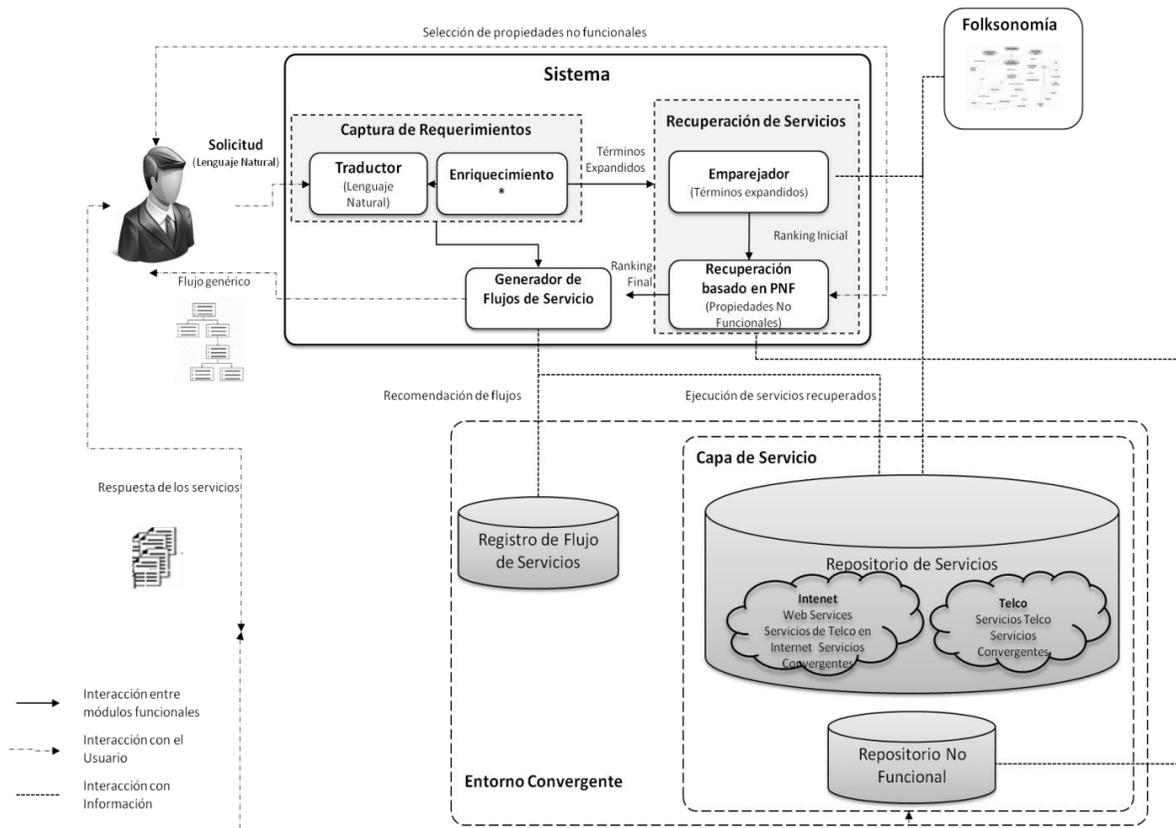


Figura 7. Sistema General de Recuperación de Servicios

#### 4.2.1.1. Selección del Emparejador

El emparejador (*Matchmaker* [6]) es el componente que utiliza el resultado del enriquecimiento semántico de la solicitud del usuario (sección 4.1.2.4) para compararlos con la descripción de los servicios, con el fin de generar un ranking inicial (ver Figura 7). Su selección se llevó a cabo considerando que fuera adaptable para el trabajo con folksonomías. De la gran cantidad de emparejadores que existen en la actualidad, la selección del emparejador que sirvió de base, fue llevada a cabo inspeccionando aquellos que han participado en encuentros donde se evalúa el rendimiento de la recuperación de servicios.

Uno de los concursos más importantes para investigadores y profesionales interesados en los temas de emparejamiento y descubrimiento semántico de servicios es el Concurso Internacional de Selección Semántica de Servicios<sup>20</sup>. A través de este concurso público, se proveen los medios necesarios para evaluar el rendimiento (precisión, tiempo de respuesta, etc.) de emparejadores semánticos de servicios Web sobre colecciones de prueba para la mayoría de formatos semánticos de servicios como OWL-S (Web Ontology Language – Semantic Markup for Web Services), WSML (Web Service Modeling Language) y SA-WSDL (Semantic Annotations for WSDL).

Tras revisar las diferentes soluciones participantes del concurso, el emparejador que se acopla perfectamente al proyecto es WSColab [33], ya que en primer lugar, está basado en el concepto de etiquetado colaborativo, al proveer mecanismos a los usuarios para realizar anotaciones que describen servicios Web con etiquetas (tags). De esta manera, como explican los autores de [33], la semántica de un servicio surge de las anotaciones de la comunidad.

<sup>20</sup> Disponible en <http://www-ags.dfki.uni-sb.de/~klusch/s3/index.html>

WSColab fue el emparejador que ganó el primer lugar en la categoría evaluación-cruzada (cross-evaluation) del concurso en el año 2009 debido a su efectividad y el corto tiempo de respuesta [150].

Con respecto al rendimiento, WSColab obtuvo un valor de 65-80% mientras que los otros sistemas tuvieron un rendimiento menor a 55-70% [33]. Adicionalmente, WSColab demostró ser el emparejador más rápido con un tiempo de respuesta promedio por solicitud por debajo de 1 milisegundo, mientras que el siguiente emparejador más rápido tenía un tiempo de respuesta de 170 milisegundos. La velocidad de un emparejador como WSColab es importante para la recuperación de servicios donde existen grandes volúmenes de los mismos, permitiendo que la atención del usuario permanezca en el sistema al transcurrir poco tiempo en mostrar resultados de progreso.

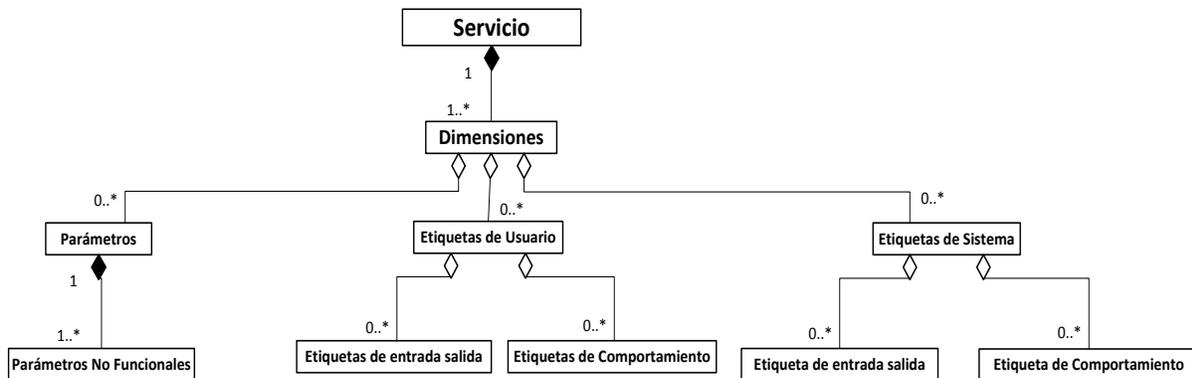
Una vez seleccionado el emparejador base del presente proyecto para la recuperación de servicios, se definieron algunas consideraciones.

- Para la búsqueda de servicios, se considera que los servicios (ver capa de servicios en la Figura 7) deben ser descritos en términos de su funcionalidad (comportamiento) e interfaz (entrada y salida), esto porque la interfaz de servicio no captura la semántica completa de funcionalidad del servicio, por lo que la funcionalidad e interfaz son complementarios para mejorar el emparejamiento de servicios [33]. Por esta razón, el sistema de recuperación de servicios está basado en dos tipos de folcsonomías: entrada/salida y comportamiento.
- Debido al posible uso de propiedades no funcionales comúnmente expresados como adjetivos (p.ej. bueno, rápido, robusto), por parte de usuarios inexpertos para buscar un servicio, se considera que criterios no funcionales cualitativos (es decir, sin valores numéricos específicos p.ej. disponibilidad, bueno, rápido) deben ser parte de la funcionalidad de servicios (comportamiento), así un servicio puede estar también descrito usando términos no funcionales (P.ej. un servicio de llamada además de estar descrito por términos como llamada, celular, comunicación, puede tener descripciones como buena, rápida, de tal manera que si un usuario solicita un servicio como: “Quiero hacer una llamada rápido”, el servicio con descripciones que contengan estas propiedades cualitativas tendrán un mayor grado de relevancia para la solicitud). El objetivo es no discriminar los términos obtenidos del usuario para la generación de un ranking inicial de servicios, sean que los términos indiquen propiedades funcionales o no funcionales.
- Obtenidos los servicios por parte del emparejador, se consideran los valores de las propiedades no funcionales de dichos servicios para realizar un proceso de recuperación con base en propiedades no funcionales (ver Figura 7).

A partir del estudio realizado al emparejador WSColab, fue posible definir dos fases dentro de la búsqueda de los servicios: clasificación de los servicios en relevantes o irrelevantes y creación de un ranking con los servicios relevantes. A continuación las fases definidas son descritas, así como las características de los servicios considerados en el proyecto.

#### **4.2.1.2. Descripción de los Servicios**

La Figura 8 introducida en [75], corresponde a una contribución de la presente propuesta de investigación para describir en detalle un servicio en el sistema de recuperación de servicios propuesto. En la figura se exponen las diferentes dimensiones que pueden ser distinguidas considerando su uso y origen: Parámetros, Etiquetas de usuario y Etiquetas de sistema.



**Figura 8. Modelo de la descripción de un Servicio**

Los servicios Web y de telecomunicaciones pueden ser descritos mediante palabras claves o etiquetas, con base en dos categorías: entrada/salida y comportamiento, tal como se observa en la Figura 8.

$$R = \{\text{nombre}, \text{archivo descriptor}, \text{etiquetas}\}$$

Donde el atributo **nombre** corresponde al nombre del servicio; el **archivo descriptor** puede ser la WSDL en caso de servicios Web o el **archivo descriptor** de despliegue de servicios que se ejecutan en entornos convergentes (como en JSLEE); y las **etiquetas** son términos que describen la funcionalidad e interfaz del servicio y se clasifican en 4 tipos:

- **Etiquetas de usuario:** son los términos generados inicialmente (P.ej. en procesos de anotación manual) ó los capturados en la fase de PLN.
- **Etiquetas de sistema:** son los términos obtenidos durante el proceso de enriquecimiento usando WordNet y Flickr.
- **Etiquetas de entrada/salida:** son las etiquetas tanto de usuario como de sistema que describen la interfaz del servicio en términos de su(s) entrada(s) y/o salida(s)
- **Etiquetas de comportamiento:** son las etiquetas de usuario y sistema que describen el comportamiento del servicio.

Los atributos más importantes de estas etiquetas son el nombre de la etiqueta y el número de usuarios que usaron ese nombre para etiquetar el servicio. Debido a que las etiquetas de sistema son producto de un enriquecimiento por parte de fuentes de conocimiento, estas no poseen un número de usuarios que las haya utilizado para etiquetar el servicio, por esta razón para este tipo de etiquetas se usa el mismo número de usuarios que el del término original que fue enriquecido.

Con el fin de evitar el problema de Inicio-Frío (Cold-Start) [129], es decir, tener servicios que no han sido descritos con etiquetas, se recurre a dos alternativas: la primera es usar servicios rastreados por motores de búsqueda que ya contienen etiquetas por defecto, como *SeekDa*<sup>21</sup> (donde ya se tiene una diferenciación entre términos de entrada/salida y de comportamiento); la segunda corresponde a descripciones de los servicios a través de anotaciones manuales (Este procedimiento es realizado por un grupo de expertos conformado por estudiantes de trabajo de grado del Grupo de Ingeniería Telemática GIT que se encuentran entre edades de 21 a 23 años y cuya investigación gira alrededor de la recuperación y composición de servicios. Los términos de entrada/salida y de comportamiento muchas veces no son discriminados y son repetidos para ambas dimensiones). Estas anotaciones son etiquetas de usuario que, una vez ejecutado el proceso de enriquecimiento, generan las etiquetas de sistema antes de la ejecución por parte del usuario.

<sup>21</sup> Disponible en <http://webservices.seekda.com>

#### 4.2.1.3. Clasificación de los servicios disponibles ofrecidos

Los términos obtenidos de la fase de PLN, luego de pasar por el proceso de enriquecimiento, son la entrada del proceso de clasificación de los servicios, donde un servicio puede ser clasificado como relevante o no, dependiendo si éste comparte términos de entrada, salida ó comportamiento en su descripción con los términos expandidos del usuario.

El siguiente es el algoritmo que explica de manera general el proceso de clasificación de servicios.

---

#### Algoritmo 1. Clasificación de Servicios

---

**INPUTS:**  $(q_{io}, q_b)$  Términos del Usuario;  $(F_{io}, F_b)$  Folcsonomías de servicios

**OUTPUTS:** *ServiciosRelevantes*

**BEGIN:**

$EtiquetasIO \leftarrow \sigma_{t_{io}eq_{io}}(F_{io})$  /\* Seleccionar Etiquetas de Servicios entrada/salida \*/

$ServicioIO \leftarrow \pi_s(EtiquetasIO)$  /\* Seleccionar Servicios de Etiquetas entrada/salida\*/

$EtiquetasB \leftarrow \sigma_{t_beq_b}(F_b)$  /\* Seleccionar Etiquetas de Servicios comportamiento\*/

$ServicioB \leftarrow \pi_s(EtiquetasB)$  /\* Seleccionar Servicios de Etiquetas comportamiento\*/

$ServiciosRelevantes \leftarrow ServicioIO \cup ServicioB$

**Return** *ServiciosRelevantes*

**END**

---

Como se definió en la Sección 3.1 (ver Figura 4), en la solicitud del usuario se pueden diferenciar términos de entrada/salida y de comportamiento. Por esta razón, se considera que la entrada para el algoritmo de clasificación de servicios del emparejador (Algoritmo 1) tiene la forma  $q_{io} - q_b$ , donde  $q_{io}$  corresponde a los términos de entrada y salida y  $q_b$  a los términos de comportamiento, ambos obtenidos a partir de la petición del usuario.

El algoritmo 1 usa dos operadores de algebra relacional para determinar si un servicio es relevante o no, estos operadores son conocidos como Selección (*Selection*) y Proyección (*Projection*). La función de Selección  $\sigma_c$ , selecciona tuplas que cumplen la condición "c", mientras que la función de Proyección  $\pi_s$ , obtiene el valor s de la tupla, sin que esta se repita, similar a la clausula SQL SELECT DISTINCT s [127].

De esta manera, se tiene un conjunto de etiquetas de entrada/salida ( $F_{io}$ ) y de comportamiento ( $F_b$ ) que describen los servicios disponibles S, donde la función  $\sigma_{t_{io}eq_{io}}(F_{io})$ , obtiene las etiquetas de la folcsonomía  $F_{io}$  que hacen parte de los términos de entrada/salida de la solicitud del usuario. Posteriormente, la función  $\pi_s(EtiquetasIO)$ , selecciona los servicios que se encuentran descritos con estas etiquetas, estos servicios corresponden a servicios relevantes identificados con los términos de entrada/salida del usuario. El mismo proceso se realiza con los términos de comportamiento. Finalmente, los servicios relevantes corresponden a la unión de los servicios *top* obtenidos con cada categoría de términos de la solicitud.

Las funciones presentadas en el algoritmo 1, fueron adaptadas de [33] con el fin de introducir las categorías de entrada/salida y comportamiento en la descripción de los recursos. A continuación son definidas para mayor claridad:

$$r(q(F_{io}, F_b)) = r(q_{io}, F_{io}) \cup r(q_b, F_b) \quad (4)$$

$$\text{Donde } r(q_{io}, F_{io}) = \pi_s(\sigma_{t_{io}eq_{io}}(F_{io})) \quad (5)$$

$$r(q_b, F_b) = \pi_s(\sigma_{t_beq_b}(F_b)) \quad (6)$$

De manera general, la ecuación 4 indica que los servicios relevantes para la solicitud  $q$  (comprendida por términos de solicitud de entrada/salida  $q_{io}$  y comportamiento  $q_b$ ), son los servicios que tienen términos comunes (tanto de entrada/salida  $t_{io}$  como de comportamiento  $t_b$ ) con los términos de la solicitud. El resultado final es la lista de servicios que cumplen con la condición que los términos obtenidos del usuario se encuentran en las etiquetas que describen dichos servicios.

#### 4.2.1.4. Ranking de los servicios relevantes

Los autores en [127], definen un ranking  $r$  de un conjunto de servicios  $S$  como una secuencia de elementos ordenados de  $S$ , es decir:  $r = (r_1, r_2, r_3, \dots, r_n)$ , donde  $n \leq |S|$ ,  $r_i \in S$ . El número  $i$  es llamado el rango (*rank*) del servicio  $r_i$  con respecto al ranking  $r$ . El ranking se genera tras la ejecución del algoritmo 2, definido a continuación:

---

#### Algoritmo 2. Ranking de Servicios Relevantes

---

**INPUTS:** ( $q_{io}, q_b$ ) Términos del Usuario ; ( $t_{io}, t_b$ ) Términos de los Servicios Relevantes; (servicios) Servicios Relevantes

**OUTPUTS:** *ServiciosRanking*

**BEGIN:**

Peso a la similitud de comportamiento asignada por el usuario  $w_b$

Peso a la similitud de entrada/salida asignada por el usuario  $w_{io}$

**for each servicio in servicios do**

    Calcular similitud Etiquetas de servicio y usuario  $sim = w_b \cdot sim(q_b, t_b) + w_{io} \cdot sim(q_{io}, t_{io})$

**if** *ServiciosRanking.isEmpty()* **then**

*ServiciosRanking.add(servicio, sim)*

**else**

        Organizar lista de mayor a menor similitud *ServiciosRanking=ordenar(servicio)*

**end if**

**end for**

**Return** *ServiciosRanking*

**END**

---

La lista de servicios clasificados como relevantes, sus etiquetas descriptoras de entrada/salida y de comportamiento y los términos obtenidos del usuario corresponden a la entrada del mecanismo para generar el ranking. Como lo indica el algoritmo 2, el rango de cada servicio es obtenido mediante la función *Sim* de similitud entre etiquetas de servicio y los términos del usuario.

#### Función de Similitud

La función de similitud total *Sim* calcula el valor final de la puntuación por servicio (el score o puntuación tiene valores de cero a uno, valores cercanos a uno indican un servicio más relevante a la solicitud) y es definida a continuación:

$$sim(q, s) = w_b \cdot sim(q_b, t_b) + w_{io} \cdot sim(q_{io}, t_{io}) \quad (7)$$

Las funciones de similitud específicas ( $sim(q_b, t_b)$  y  $sim(q_{io}, t_{io})$ ) entre la solicitud  $q$  y los servicios  $S$ , se definieron basándose en un modelo muy popular en los sistemas de recuperación de información, llamado modelo de espacios vectoriales (VSM) [128]. Este modelo permite realizar búsqueda de documentos que contienen términos de una solicitud. Su nombre se debe a que los documentos y solicitudes son

representados como vectores de términos, mostrando cada término en el vector con un peso dentro del documento. Usualmente los términos son palabras y sus valores en el vector son ponderados de acuerdo a determinados esquemas. El esquema de ponderación más común es el que considera pesos locales (es decir, le da más valor a la palabras que aparecen varias veces en el documento) y globales (es decir, la repetición de la palabra en otros documentos, dándole mayor valor a las palabras que compartan menos documentos). Este esquema de ponderación se conoce comúnmente como ponderación *tf-idf* (term frequency-inverse document frequency), el cual se describe posteriormente en esta sección.

Las constantes  $w_b$  y  $w_{io}$  de la ecuación 7 son los pesos asignados para calcular la similitud de términos de usuario con el servicio para las descripciones de comportamiento y entrada/salida respectivamente, los cuales pueden tomar valores entre 0 y 1. La definición de estos pesos permite calcular el aporte individual de cada categoría, y su suma debe ser igual a 1.

La ecuación 7 da mayor relevancia a los servicios que comparten tanto términos de entrada/salida y de comportamiento con los términos de la solicitud del usuario y da una menor relevancia a los que solo comparten términos en una sola dimensión ya sea de entrada/salida o de comportamiento.

A través de un bucle de repetición (loop), el algoritmo 2 calcula la función de similitud para cada uno de los servicios relevantes y tiene dos posibles flujos de lógica: primero, si la lista de los servicios en el ranking esté vacía, agrega el primer servicio relevante del bucle a dicha lista; segundo, con la función *ordenar* reordena la lista de los servicios de mayor a menor, de acuerdo al puntaje obtenido a partir de la función de similitud. Finalmente, una vez acabado el bucle, se obtiene el ranking generado.

### Ponderación *tf-idf*

La forma de calcular la ponderación *tf-idf* es a través del producto de los términos TF e IDF. TF ó *Term Frequency* (frecuencia del término), es el número de veces que un término ocurre en un documento. Por otro lado IDF ó *Inverse Document Frequency* (Frecuencia de documento inversa) considera que los términos que se repiten en varios documentos son menos informativos que términos raros que ocurren en unos pocos, es decir, asigna mayor valor a términos que aparecen en pocos documentos. El esquema de ponderación *tf-idf* para un término  $t$  en un documento  $d$  es calculado de la siguiente manera:

$$w(t, d) = tf_{t,d} * idf_t \quad (8)$$

Así,  $w(t,d)$  es alto cuando el término  $t$  ocurre muchas veces en un conjunto pequeño de documentos; es bajo cuando el término  $t$  ocurre pocas veces en un documento o cuando ocurre en muchos documentos; y es muy bajo cuando ocurre en casi todos los documentos.

De acuerdo a lo anterior, la representación vectorial del documento es la siguiente:

$$d_j = (w(t_1, d_j), w(t_2, d_j), \dots, w(t_n, d_j)) \quad (9)$$

Y la puntuación del documento respecto a una solicitud es definida como la sumatoria de las ponderaciones *tf-idf* en el documento, considerando los términos usados en la solicitud:

$$score(q, d) = \sum_{t \in q} w(t, d) \quad (10)$$

En el modelo, un vector puede ser normalizado dividiéndolo con la longitud del mismo, de esta manera la puntuación normalizada sería la definida en la ecuación 11:

$$score(q, d) = \frac{\sum_{t \in q} w(t, d)}{\sqrt{\sum w_i^2}} \quad (11)$$

Basándose en el modelo de espacio vectorial anteriormente descrito, fue definida la función de similitud como:

$$sim(qb, tb) = \frac{\sum_{t \in qb} W_{s,t}}{\sqrt{\sum_{t \in qb} W_{s,t}^2}} \quad (12)$$

La ecuación 12 hace referencia a la función de similitud entre los términos de comportamiento del usuario y las descripciones de los servicios, donde el valor de  $W_{s,t}$  sigue un esquema de ponderación *tf-idf* equivalente al definido en el emparejador [33], descrito a continuación:

$$W_{s,t} = tf_{s,t} \cdot idf_t \quad (13)$$

Donde,

$$tf_{s,t} = \frac{n_{s,t}}{N_t} \quad (14)$$

$$idf_t = \log \frac{|S|}{1 + |S_t|} \quad (15)$$

La expresión  $tf_{s,t}$  (frecuencia de la etiqueta), se calcula como la razón entre el número de usuarios que usaron la etiqueta  $t$  para describir el comportamiento del servicio ( $n_{s,t}$ ) con el número de etiquetas que todos los usuarios usaron para la descripción del comportamiento del servicio ( $N_t$ ). En otras palabras, son de mayor relevancia en el ranking aquellos servicios que tienen etiquetas que muchos usuarios usaron para describirlo. La expresión  $idf_t$  (Frecuencia de documento inversa) relaciona el número de todos los servicios registrados ( $S$ ) con el número de los servicios que tienen la etiqueta  $t$  que lo describen ( $S_t$ ). De tal forma que se dará mayor relevancia a los servicios que tienen etiquetas usadas para describir pocos servicios.

Finalmente, el cálculo de la similitud en la ecuación 12 es normalizado con la longitud del vector. Este valor permite que la cantidad de etiquetas usadas para describir un servicio pueda ser muy grande sin alterar el cálculo de la similitud. En el ranking resultante, se obtiene un valor numérico de 0 a 1 para cada uno de los servicios relevantes, entre más cercano a 1, es más próximo a la solicitud del usuario.

### 4.3. RECUPERACIÓN DE SERVICIOS CON BASE EN PROPIEDADES NO FUNCIONALES

Tal como se concluyó en la Sección 2.2.1.2, dentro del proceso de recuperación de servicios, es necesario tener en cuenta las *Propiedades no Funcionales* (PNF) de los mismos, dado que esto incrementa el nivel de satisfacción percibido por los usuarios al recuperar servicios, brindándole mayor calidad y ajustándolo más a sus necesidades. Estas propiedades permiten calificar la operación de un sistema, más allá de su comportamiento específico.

#### 4.3.1. Propiedades No Funcionales (PNF).

El desarrollo de aplicaciones que involucren propiedades funcionales ha sido ampliamente discutido en diferentes escenarios. Sin embargo, la inclusión de aquellos parámetros que no conciernen a características de comportamiento de servicios, no han sido lo suficientemente exploradas para comprender los beneficios que estos conllevan. Es por ello que las PNF, son relevantes en los procesos de recuperación de servicios, al considerar criterios de calidad específicos que los usuarios expresan en sus requerimientos.

Frecuentemente las PNF de un servicio se denominan cualidades de un sistema, o simplemente objetivos de calidad y pueden dividirse en dos categorías principales, presentadas a continuación [132]:

1. *Cualidades de ejecución*: seguridad y reusabilidad. Son parámetros medibles y observables en tiempo de ejecución.
2. *Cualidades de evolución*: capacidad de prueba, mantenimiento, extensibilidad y escalabilidad. Se encuentran incorporados en la estructura estática del software del sistema.

Como elementos requeridos en esta etapa de procesamiento, es necesario en primera instancia, considerar el valor que tiene para los usuarios algún parámetro no funcional de un servicio. Cabe destacar que los usuarios objetivo, son personas inexpertas con poco o ningún conocimiento sobre tecnologías de Internet (Propiedades Funcionales y No Funcionales de los Servicios), lo que propicia el uso de un proceso asistido de selección de atributos no funcionales por parte de los usuarios finales. Esta estrategia se detalla a continuación.

#### **4.3.2. Selección de parámetros No Funcionales de los servicios.**

En esta instancia, con el fin de desarrollar la Recuperación de Servicios con base en PNF es necesario identificar y precisar las propiedades y sus valores respectivos, tanto en servicios Web como de Telecomunicaciones.

El estudio realizado sobre los esfuerzos adelantados en Recuperación de Servicios considerando sus PNF, reveló cuáles propiedades son las más relevantes a la hora de ser presentadas a los usuarios. Existen algunos repositorios de servicios que son descritos con propiedades no funcionales, como Seekda. Éste último provee un gran directorio de servicios etiquetados con algunas PNF, tales como: documentación y disponibilidad. Por otro lado, se tiene el repositorio de QWS Dataset<sup>22</sup> que aunque contiene un número menor de servicios Web, en contraste a Seekda, considera una mayor cantidad de PNF para cada servicio (Tiempo de respuesta, Rendimiento, Habilidad de éxito, Confiabilidad, Cumplimiento, Mejores Prácticas y Latencia).

Como se argumentó en la Sección 2.2.1.2, es pertinente considerar PNF para los servicios de telecomunicaciones. En este sentido, las PNF que deben cumplir se encuentran en los lineamientos de las diferentes empresas del sector o en la documentación ofrecida por software donde los servicios son ejecutados (entorno de ejecución). Para el caso del presente proyecto, se tuvieron en cuenta algunas de las propiedades que especifican entornos de ejecución de servicios como *Rhino Slee*<sup>23</sup>, entre las cuales se encuentran disponibilidad, rendimiento, tiempo de respuesta, entre otros. Los nombres y valores de las propiedades no funcionales de los servicios se almacenaron en una base de datos relacional descrita en el **Anexo C**.

#### **4.3.3. Ranking de Servicios con PNF.**

Posterior a la selección de servicios a partir de elementos capturados de la solicitud del usuario (ranking inicial), es preciso identificar cuál de esos servicios, en términos de parámetros no funcionales, se ajusta a los requerimientos del usuario. Para lograr esto, se genera un ranking con base en los valores de las propiedades no funcionales de los servicios. A continuación se hace una descripción detallada de los elementos considerados dentro del proceso de restructuración del ranking de servicios inicial.

##### **4.3.3.1. Factor de relevancia.**

Cuando un usuario selecciona una PNF, deja por fuera otras propiedades que probablemente posean igual o mayor grado de importancia para la recuperación de servicios, comprometiendo el ranking final. Para superar estos inconvenientes, y con el fin de brindar mayor soporte y flexibilidad al usuario final, a continuación se presenta la estrategia seguida, la cual busca asegurar que todas las propiedades no

---

<sup>22</sup> Disponible en: <http://www.uoguelph.ca/~qmahmoud/qws/index.html>

<sup>23</sup> Disponible en: <https://developer.opencloud.com/devportal/display/OCDEV/Rhino+2.3>

funcionales sean consideradas mediante la asignación de unos pesos durante el establecimiento del ranking final de servicios.

La forma para garantizar este funcionamiento se muestra a través de la ecuación 16. Inicialmente en esta ecuación fueron definidos dos grupos de propiedades, denominadas  $h(x)$  *hard selection* y  $s(x)$ , *soft selection*. Esta definición se basó en la clasificación de criterios no funcionales para la selección de servicios Web, establecidos en [133]. Por un lado,  $h(x)$  representa aquellos parámetros no funcionales que han sido seleccionados por el usuario que tendrán un mayor peso respecto a los que no fueron seleccionados. Por otra parte,  $s(x)$  representa los parámetros no seleccionados.

Así mismo en la ecuación 16, se definen dos variables:  $T$  y  $x$ , las cuales representan el total de PNF y el número de propiedades seleccionadas respectivamente. Existe adicionalmente una variable  $k$ , que simboliza el peso asignado tanto a  $h(x)$  como a  $s(x)$ , en el caso que el número de parámetros seleccionados sea diferente de cero y del total.

En este mismo sentido, se ha propuesto un factor denominado de relevancia ( $R$ ), cuyos valores oscilan entre cero y uno, y que además tiene como propósito establecer una relación entre el peso que podrían poseer las PNF seleccionadas y las no seleccionadas. Es decir, entre más cercano a uno, el valor de las PNF seleccionadas será mayor, mientras que para valores próximos a cero, el valor de las PNF seleccionadas será menor.

Particularmente, puede ocurrir que un usuario no seleccione ninguna PNF, para lo cual el sistema genera un ranking con todas las PNF. Para este caso, es tenido en cuenta otro factor (afinación), expuesto en detalle a continuación, el cual relaciona la pertinencia considerada por un usuario con relación a las PNF.

- $s(x) = \text{soft selection}$
- $h(x) = \text{hard selection}$
- $n = \text{número total de PNF.}$

$$W_i = \left\{ \begin{array}{ll} \text{si} & x = T \text{ o } x = 0 \\ \text{otro caso} & k = \left( \left( \frac{T}{x} - 1 \right) * R + 1 \right) * \left( \frac{x}{T} \right) \end{array} \right. \left. \begin{array}{ll} s(x) = 0 & h(x) = \frac{1}{T} \\ s(x) = \frac{1-k}{T-x} & h(x) = \frac{k}{x} \end{array} \right\} \quad (16)$$

Cabe notar que una vez obtenido el valor del peso marcado por  $W_i$ , la sumatoria de cada uno de esos pesos debe corresponder a la unidad, como se muestra en la ecuación 17.

$$1 = \sum_{i=0}^n W_i \quad (17)$$

#### 4.3.3.2. Ponderación de PNF de los servicios seleccionados.

En esta sección se muestra como a partir del conjunto de servicios obtenidos del ranking inicial, se realiza una reasignación de las posiciones de los servicios que lo conforman con base en sus parámetros no funcionales. De esta forma, se genera un nuevo ranking de acuerdo al tipo de PNF considerada. Los autores de [63] definen una variable  $E$ , que establece el peso para cada propiedad, tal como se observa en las ecuaciones 18, 19 y 20.

Para el caso en el cual el tipo de PNF sea numérico, la ecuación 18 muestra el valor de  $E$ , que depende de dos valores un valor máximo ( $V_{max}$ ) y un valor mínimo ( $V_{min}$ ) correspondientes a las magnitudes que puede tomar cada propiedad. Adicionalmente se consideran dos casos, el primero donde se desea que la

propiedad tenga una máxima magnitud (P.ej. Disponibilidad) y el segundo caso, donde el servicio va a ser mejor si el valor de la propiedad no funcional es mínimo (P.ej. latencia). Así, el valor  $v$ , mostrado en la misma ecuación, representa el valor específico de cada parámetro para cada uno de los servicios que están siendo evaluados.

$$E = \begin{cases} 1 - \left( \frac{V_{max}-V}{V_{max}-V_{min}} \right) \\ \left( \frac{V_{max}-V}{V_{max}-V_{min}} \right) \end{cases} \quad (17)$$

Si el parámetro  $E$  es de tipo booleano, los autores de [63] determinan que su valor será de uno (1) si coincide con el valor deseado o cero (0) si es contrario a lo esperado.

$$E = \begin{cases} 1 \text{ Si el criterio es conocido} \\ 0 \text{ en otro caso} \end{cases} \quad (18)$$

Finalmente, si el tipo de la propiedad es un *String* o cadena de caracteres, se comprueba si dentro de la propiedad existe un conjunto de caracteres que coincida con el valor deseado, y así calcular el porcentaje de la descripción total de la propiedad representada dentro de la descripción esperada (Ecuación 20).

$$E = \left\{ \frac{e_1+e_2+\dots+e_n}{n} \right\} \quad (19)$$

Los valores que puede tomar  $E$  están comprendidos en el rango de 0 a 1 y representan cuán buena es la propiedad no funcional en cada uno de los servicios, respecto a un valor óptimo deseado.

Tras el cálculo de  $E$  para cada propiedad no funcional del servicio, se estima el valor total para su posición en el ranking. Este total se denomina NF:

$$NF = \left\{ (W_1E_1^r + W_2E_2^r + \dots + W_nE_n^r)^{\frac{1}{r}} \right\} \quad (20)$$

La ecuación 21, es construida a partir de un método cuantitativo tomado de una técnica de calificación LSP (*Logic Scoring Preferences*) [64], la cual fue creada originalmente para tomar decisiones sobre problemas de múltiples criterios, como la selección de hardware. Esta técnica fue adaptada con el fin de reflejar relaciones de dependencia entre preferencias no funcionales, para finalmente poder establecer un criterio de selección de servicios con propiedades no funcionales relevantes. Adicionalmente, los autores en [63] proponen una función que transforma un conjunto de valores a una escala normalizada que indica el grado de satisfacción de la correspondiente preferencia. Luego, todos los valores de preferencia son ordenados a través del uso de técnicas de agrupación de estructuras escalonadas, para construir una preferencia global, como lo es requerido en el presente trabajo. Todo esto se logra mediante una función de preferencia de agregación, denominada *generalización conjunción/disjunción* o *And/Or*, la cual a través de la ponderación de medias, obtiene la preferencia global [63].

Adicionalmente, la ecuación 21 representa una sumatoria, donde  $W$  es obtenido de acuerdo a lo señalado en la ecuación 16 y  $E$  depende del tipo de PNF que está siendo procesada dentro de la generación del ranking. De igual manera, la variable  $r$  representa el grado de simultaneidad o relación de un grupo de características dentro de un bloque de agregación. Esta variable  $r$ , obtenida a partir de un cálculo de lógica dinámica establecido en [63] con valores entre cero y uno, está definida como un número real descrito en términos de conjunción y disjunción, calculado de acuerdo a la relación de conjunción que contienen los parámetros objeto de evaluación.

Cabe resaltar que conjunción, hace referencia a cuán deseable es que un conjunto de características estén juntas, dentro de un bloque de agregación, mientras que disyunción representa el caso contrario. En otras palabras y de acuerdo a la ecuación 21, si el valor de  $r$  tiende a uno, la función evidenciaría un comportamiento disyuntivo, en caso contrario cuando este valor se aproxime a cero, se tornaría una función de conjunción [134].

#### **4.4. OBTENCIÓN FINAL DEL RANKING DE SERVICIOS**

El resultado final de la lista de servicios expuesto a los usuarios, considera tanto la recuperación de servicios con base en propiedades funcionales, como propiedades no funcionales, esto con el objetivo de obtener una respuesta más cercana a la petición del usuario.

##### **4.4.1. Factor de Afinación.**

El ranking final de servicios utiliza el ranking inicial  $F$  obtenido en la sección 4.2 (Ver ecuación 7), el cual permitió reducir el espacio de búsqueda de los servicios, adicionalmente considera el análisis de PNF de los servicios, expresado en la variable  $NF$  (Ver ecuación 21). De esta manera, se define en la ecuación 22 un ranking final  $R$  para cada servicio.

Adicionalmente, se ha definido el factor de afinación  $\alpha$ , cuyo valor se encuentra en el intervalo entre cero y uno, simbolizando el valor que un usuario otorga (mayor o menor) al ranking inicial respecto al ranking con propiedades no funcionales. Para garantizar que la proporción de ambas sea igual a la unidad, el factor  $(1-\alpha)$  es multiplicado por el valor obtenido con las PNF, mientras que  $\alpha$  es multiplicado por  $F$ , representando el valor asignado al ranking inicial.

$$R = \alpha F + (1-\alpha)NF \quad (21)$$

Finalmente, la variable  $R$  contiene el peso final asignado a un determinado servicio, teniendo en cuenta tanto la evaluación de propiedades funcionales, como no funcionales. Esta ecuación representa la ponderación final dada a un determinado servicio, la cual define su posición en el ranking presentado al usuario final.

##### **4.4.2. Flujo de control genérico.**

Como valor agregado a los objetivos planteados en el presente trabajo, se encuentra la inclusión de un flujo de control que indica el orden en el cual los servicios recuperados deberían ejecutarse. Existen algunos trabajos relacionados que han estimado la inclusión de un flujo a partir de plantillas obtenidas con base en las peticiones ingresadas por los usuarios [55]. Si bien esto facilita el establecimiento del flujo de control de los servicios, genera limitaciones y restricciones a los usuarios, ya que se concentra en solicitudes de mayor complejidad, las cuales requieren el uso de un lenguaje formal y técnico.

Por otra parte, existen otros enfoques que establecen un flujo de composición a partir de las interfaces de los servicios [133] (elementos de entrada, salida). Este tipo de aproximaciones son de gran importancia dentro de actividades como la composición de servicios, sin embargo, se encuentra fuera del alcance definido por el presente proyecto.

Dando respuesta a la posibilidad de incluir un flujo de control de acuerdo a la solicitud declarada por los usuarios, se exponen a continuación los elementos a tener en cuenta para la identificación de aquellos parámetros de control, que permitan directamente obtener una estructura que represente el flujo de control implícito en la solicitud de los usuarios.

#### 4.4.2.1. Identificación de parámetros de control

A nivel de lenguaje natural, existen diversas palabras que indican un orden o una secuencia de elementos dentro de una frase (palabras ordinales, conjunciones, etc.). Es a partir de la captura de estos términos, donde surge la posibilidad de identificar y establecer una secuencia de elementos dentro de una oración. La identificación de palabras, como por ejemplo: *y, o, sin embargo, después, antes, sí, primero*, etc., permite segmentar una oración y de este modo establecer secciones de palabras con las que se podría identificar la forma como se quiere ejecutar una solicitud determinada.

La identificación de los parámetros de control se logra a partir de una definición previa de una lista de palabras. Posterior a ello, mediante un análisis sintáctico de la solicitud, se puede establecer si alguna de las palabras listadas se encuentra inmersa dentro de una petición. Si el proceso de identificación determina que existen palabras de control dentro de la solicitud, se procede a separar los elementos de acuerdo a la palabra identificada. Este conjunto de elementos se denominan *bloques* y están constituidos por un grupo de palabras que contienen propiedades funcionales de los servicios que se quiere identificar. De manera general, cuando la identificación de palabras de control se concentra en conjunciones, el sistema contendrá un número  $N+1$  de bloques, donde  $N$  representará el número total de palabras de control identificadas.

#### 4.4.2.2. Generación del flujo de control.

A partir de las palabras de flujo de control identificadas, se establece un grafo de bloques<sup>24</sup> vacíos, destinados a contener los servicios recuperados. Para esto, se emplea un software que permite visualizar gráficos; de tal manera que el grafo puede desplegarse facilitando la observación del flujo de servicios creado.

Dentro de las estructuras comunes de un flujo de control, se pueden encontrar patrones que contienen relaciones entre sus módulos constituyentes, estas relaciones de acuerdo a lo expuesto en [136] están definidas como:

- *Estructuras secuenciales*: bloques adyacentes, dispuestos uno a continuación del otro.
- *Estructuras paralelas*: bloques concurrentes, que se ejecutan al tiempo.
- *Estructuras Condicionales*: bloques que según la evaluación de una expresión lógica (una condición), se realiza una opción u otra.

Inicialmente con las palabras de control, es posible definir relaciones como las mencionadas anteriormente, lo que lleva al establecimiento de patrones para determinadas palabras. Por ejemplo, si la palabra de control identificada es “y” entonces esto llevaría a que dos servicios se ejecuten paralelamente, lo que se traduce a nivel del grafo como dos bloques secuenciales paralelos.

Una vez terminado el grafo que contiene el conjunto de bloques ordenados a partir de las palabras de control identificadas, se procede a la integración con el ranking de servicios introducido en la sección 4.4. Como se señaló, un bloque de control está constituido por un conjunto de palabras, de las cuales se obtiene un ranking de servicios, por lo que es necesario establecer nuevamente la relación entre el ranking y el bloque de control dentro del grafo, para entregar al usuario el flujo de control genérico final.

En la actualidad existe una cantidad considerable de herramientas para la generación de grafos; Éstas varían dependiendo del lenguaje de programación empleado para su desarrollo y de sus licencias de uso: *OpenSource*, privativas, etc. Posterior al análisis detallado de herramientas software para la visualización de gráficos, se obtuvo que la herramienta *Graphviz*<sup>25</sup> brinda diversas facilidades para generar un grafo con

---

<sup>24</sup> Estos bloques corresponden a los nodos de un grafo y consecuentemente, las aristas entre ellos establecen sus relaciones, lo que permite definir estructuras de flujo de control comunes (patrones).

<sup>25</sup> Disponible en: <http://www.graphviz.org/>

condiciones óptimas y adecuadas, debido a su simpleza en el desarrollo a través del lenguaje de descripción de texto plano DOT, y al API que ofrece para desarrollos en lenguaje JAVA.

Finalmente, a partir del grafo obtenido se procede a generar un identificador del flujo, el cual es almacenado en un repositorio (**Anexo B**). Adicionalmente, las palabras que identifican a los servicios recuperados, son almacenadas con el identificador del gráfico del flujo (Representación gráfica del grafo final). Esto se realiza con el fin de agilizar el proceso de recuperación, para futuras solicitudes, es decir, dado el evento en el cual un usuario introduzca una solicitud similar a otra realizada anteriormente, el sistema podrá a través de la consulta del repositorio, recomendar flujos finales de servicios obtenidos previamente, evitando ejecutar los procesos de recuperación señalados en las secciones anteriores. Este procedimiento se explica en detalle a continuación.

#### 4.4.3. Recomendación del Flujo de Control genérico.

La recomendación del Flujo de Control genérico agiliza la presentación de resultados a los usuarios sin necesidad de un exhaustivo procesamiento. Esto se realiza a través del uso de antecedentes o historial de otros usuarios. El presente trabajo pretende brindar una alternativa de recomendación a los usuarios, sin que esto implique necesariamente que el usuario tenga que seleccionar la recomendación que se le presenta, ya que de manera concurrente se ejecuta el proceso de recuperación de servicios.

Cuando un usuario ingresa una petición, esta pasa por un proceso de análisis de lenguaje natural, uno de esos procesos es la *Tokenización* (descrito en la sección 3.1.2.2), a través del cual se obtienen como salida los elementos básicos (*tokens*) de la solicitud. Este conjunto de tokens es comparado con los términos que están asociados a los diferentes flujos almacenados de solicitudes previas, con el fin de detectar coincidencias y realizar el proceso de recomendación.

En este sentido, para realizar la recomendación de flujos relevantes a los tokens de la solicitud, se definen tres vectores  $Q$ ,  $q$  y  $f$ , donde  $Q$  representa el conjunto de tokens recuperados de la solicitud,  $q$  el vector que representa con unos y ceros el vector  $Q$  y  $f$  el vector que relaciona los tokens del usuario con los términos asociados al flujo almacenado. A continuación se define cada uno:

$$\begin{aligned} Q &= \{Q_1, Q_2, Q_3, \dots, Q_i, \dots, Q_n\} \\ q &= \{q_1, q_2, q_3, \dots, q_i, \dots, q_m\} \\ f &= \{f_1, f_2, f_3, \dots, f_i, \dots, f_m\} \end{aligned} \quad (22)$$

donde  $Q_i =$  Token recuperado de la solicitud en la posición  $i$

$$q_i = \begin{cases} 1 & i \leq n \\ 0 & i > n \end{cases} \quad (23)$$

$$f_i = \begin{cases} 1 & \text{Si } Q_i \text{ está asociado al flujo} \\ 0 & \text{Si } Q_i \text{ NO está asociado al flujo} \\ 1 & i > n \end{cases} \quad (24)$$

- $n =$  número de tokens recuperados de la solicitud
- $m =$  número de términos asociados al flujo más el número de tokens de la solicitud que no se encuentran en el flujo

Para brindar una mayor comprensión de los vectores definidos anteriormente, se considera un ejemplo donde se tiene un conjunto de tokens recuperados de la solicitud  $Q=\{\text{llamada, celular, amigo}\}$ , adicionalmente en el registro de flujos de servicios almacenados, existe un flujo con los términos asociados

“llamada, teléfono, bueno, sms”, el vector  $f$  para este flujo de servicios sería  $f=\{1,0,0,1,1,1\}$  y el vector  $q$  que representa el vector  $Q$  en términos numéricos corresponde a  $q=\{1,1,1,0,0,0\}$ .

Con el fin de dar mayor relevancia a los flujos que contienen más términos en común con la solicitud y menos términos adicionales que pueden no estar relacionados, es necesario definir una métrica que permita identificar la similitud de los vectores  $q$  y  $f$ . Actualmente, existe una gran variedad de medidas de similitud y distancia entre vectores, ampliamente usados en sistemas de recuperación de información, máquinas de búsqueda, sistemas de pregunta–respuesta, entre otros. Entre las más populares y aceptadas se encuentra la similitud de cosenos, usada para calcular la similitud entre los vectores  $q$  y  $f$ . Su utilidad se debe a que mide la similitud entre dos vectores de acuerdo al ángulo entre ellos y no por su distancia, ya que se ha comprobado que medidas basadas en distancia como la *distancia Euclídea* no indican claramente la similitud entre dos vectores [137]. Esto se debe, entre otras cosas, a que la distancia es mayor para vectores con diferentes longitudes así posean componentes similares.

De esta manera, como se muestra en la figura 9, esta métrica determina el valor máximo de similitud entre vectores que posean el mismo ángulo. De igual forma, los valores decrecientes de similitud dependen del ángulo formado por los vectores que se están comparando, ya que, la función coseno es monóticamente decreciente en el intervalo 0 a 180 grados.

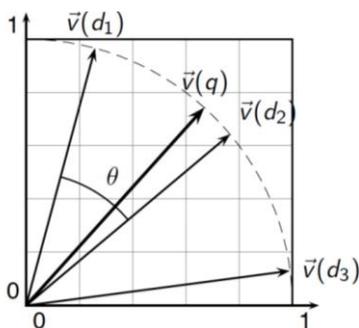


Figura 9. Similitud de Vectores.

Para determinar el nivel de semejanza entre los vectores  $q$  y  $f$ , se calcula la *similitud de cosenos* según la siguiente ecuación:

$$\cos(\vec{q}, \vec{f}) = \frac{\vec{q} \cdot \vec{f}}{|\vec{q}| |\vec{f}|} = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{f}}{|\vec{f}|} = \frac{\sum q_i f_i}{\sqrt{\sum q_i^2} \sqrt{\sum f_i^2}} \tag{25}$$

Aplicando la métrica de similitud expuesta al presente trabajo, se tiene que: por cada flujo almacenado existe un par de vectores que representan la solicitud y el flujo; aplicando la ecuación 26, se calcula el nivel de similitud de estos vectores, los cuales a su vez, son organizados de forma descendente, con el fin de mostrar los dos flujos con mayores valores de similitud. En caso que la recomendación sea aceptada por el usuario, el proceso de recuperación que se lleva a cabo paralelamente se detendrá y la recomendación se adoptará como resultado, mostrando la respuesta al usuario y ejecutando los servicios del flujo; es de resaltar que la ejecución de cada servicio es independiente del resultado de sus antecesores, ya que esto no se encuentra dentro del alcance del presente trabajo. En el caso de que la recomendación sea rechazada, el proceso paralelo de recuperación continúa su curso normal hasta generar un nuevo flujo de servicios.

## 4.5. EJEMPLO

En esta sección se presenta un estudio de caso que permite visualizar las funciones que soportan la recuperación de servicios. Para este ejemplo, se considera como entrada al módulo, un resultado similar al obtenido del ejemplo de la fase de procesamiento de lenguaje natural expuesto en la Sección 3.3:

“Bloque1: **Recibir:** funcional-comportamiento, **veinte:** funcional-entrada/salid, **reporte:** funcional-comportamiento, **tráfico:** funcional-comportamiento.  
**después:** Control

Bloque 2: **reunión:** funcional-comportamiento, **audio:** funcional-comportamiento, **decisión:** funcional-comportamiento”

Donde un conjunto de palabras clave con una serie de etiquetas asociadas, ingresa al modulo de enriquecimiento semántico definido en la Sección 4.1. El módulo propuesto utiliza dos fuentes de conocimiento WordNet y Flickr; para ejemplificar lo ocurrido en este proceso se toma la palabra “*reporte*” y a la fuente de conocimiento WordNet, la cual brinda un conjunto de palabras semánticamente relacionadas, enriqueciendo el significado de la palabra, como se muestra a continuación:

“**Reporte** → *historia, estudio, cuenta, composición, artículo*”<sup>26</sup>

Este conjunto de palabras se obtienen de acuerdo al conjunto de relaciones semánticas (*hiperónimos, hipónimo, sinónimos* etc.) internas que posee WordNet. El siguiente paso es la generación del ranking inicial de servicios, en esta etapa se tienen en cuenta las etiquetas funcionales (comportamiento, entrada/salida) obtenidas en la Sección 3.3, ya que el repositorio de servicios (ver figura 8) contiene descripciones que relacionan directamente la funcionalidad de los mismos. Teniendo esto en cuenta y ejecutando cada uno de los algoritmos señalados en la sección 4.2., se realiza un proceso de emparejamiento entre las etiquetas descriptoras de servicio y las etiquetas de los términos de entrada, arrojando como resultado lo siguiente (para el ejemplo se muestran los primeros 6 servicios relevantes de los dos bloques de solicitudes utilizados):

**Tabla 5. Salida proceso de recuperación inicial de servicios.**

Bloque 1	Bloque 2
1. ReportingService	1. HeaderTimeService
2. QueueService	2. CommitteeMeetingService
3. DealService_v1	3. StrikeIronRealTimeQuery
4. PostItService	4. Perceval_VoIP_manage
5. TopLabService	5. PostItService
6. USWeather	6. AWSECommerceService

Los nombres de los servicios están dados, tal como se muestran en el repositorio, ya que fueron tomados de diferentes recursos en la web. Cabe notar que los usuarios finales, si lo desean, pueden seleccionar de entre un conjunto de nueve propiedades no funcionales disponibles; suponiendo que el usuario del sistema selecciona todas las propiedades no funcionales, el sistema a través de lo señalado en la sección 4.3 presenta un nuevo ranking como el siguiente:

<sup>26</sup> Disponible en:

<http://wordnetweb.princeton.edu/perl/webwn?s=report&sub=Search+WordNet&o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&h=000000>

**Tabla 6. Salida proceso de recuperación de servicios con parámetros no funcionales.**

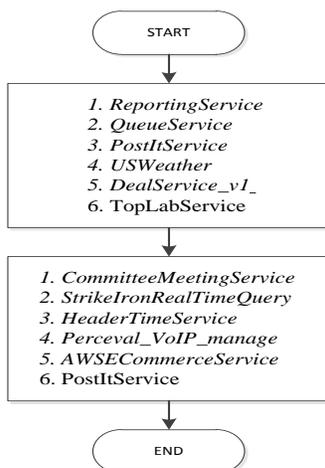
Bloque 1	Bloque 2
1. QueueService	1. Perceval_VoIP_manage
2. ReportingService	2. CommitteeMeetingService
3. PostItService	3. StrikeIronRealTimeQuery
4. USWeather	4. AWSECommerceService
5. DealService_v1	5. HeaderTimeService
6. TopLabService	6. PostItService

Una vez se tienen estos dos rankings, se procede a la obtención del ranking final, tal como se señala en la sección 4.4. Este proceso tiene en cuenta un *factor de afinación* (sección 4.4.1) el cual se estableció en 0.8, con el propósito de brindar mayor peso al ranking inicial que al ranking de propiedades no funcionales. El ranking final obtenido se muestra a continuación.

**Tabla 7. Salida final proceso de recuperación de servicios.**

Bloque 1	Bloque 2
1. ReportingService	1. CommitteeMeetingService
2. QueueService	2. StrikeIronRealTimeQuery
3. PostItService	3. HeaderTimeService
4. USWeather	4. Perceval_VoIP_manage
5. DealService_v1	5. AWSECommerceService
6. TopLabService	6. PostItService

Adicionalmente, el sistema a partir de las palabras identificadas con la etiqueta control, genera la estructura de control que determinará el flujo de servicios. Como la palabra identificada con esta etiqueta fue el término “*después*”, el sistema primero mostrará un flujo con N+1 bloques, donde N representa el número de palabras de control identificadas, para el actual ejemplo N es igual a uno. Segundo, el sistema mostrará dos bloques secuenciales ya que el término *después*, significa que un proceso (bloque) deberá ir tras el otro. El resultado final mostrado a los usuarios es por tanto como el que se indica a continuación:



**Figura 10. Flujo final de servicios.**

## 4.6. RESUMEN

Este capítulo, corresponde a una contextualización de las diferentes técnicas de selección de servicios, basadas en un emparejamiento sintáctico de etiquetas, complementadas a su vez por un enriquecimiento semántico de las mismas. Es de resaltar la contribución que se realiza en este trabajo a los tradicionales procesos de recuperación, con la inclusión de un proceso de enriquecimiento semántico, ya que con ello se amplía, diversifica y facilita el análisis del lenguaje usado. Adicionalmente, como se hizo evidente en la sección 2.2.3, la complejidad requerida para ingresar solicitudes en los sistemas de recuperación de servicios es alta, razón por la cual se busca simplificar la forma como un usuario puede realizar su petición al sistema. Gracias a la inclusión de palabras relacionadas semánticamente a través de ontologías y de clústeres, es posible obtener un mayor nivel de comprensión de la solicitud, alcanzando mejor precisión y eficacia en el análisis.

Por otro lado, la adición de propiedades no funcionales en el proceso de selección de servicios, aporta nuevos elementos en la recuperación, obteniendo servicios de mayor calidad, mejor desempeño y más fiabilidad para los usuarios; lo que conlleva también al incremento de la satisfacción de los usuarios finales, al obtener servicios con mejores características no funcionales.

Adicionalmente, en el capítulo se define una aproximación de la generación de un flujo de servicios a partir de la solicitud del usuario. Este flujo se obtiene de acuerdo al análisis de palabras de control, que hacen parte de las peticiones de los usuarios.

El capítulo finaliza con un ejemplo que muestra los procesos y las salidas de cada una de las tareas que soportan la recuperación de servicios. Inicialmente se muestra el ranking obtenido a partir de los términos de entrada (Sección 3.3), seguido se genera un nuevo ranking haciendo uso de las propiedades no funcionales de cada servicio y posteriormente se obtiene el ranking final que tiene en cuenta los dos rankings mencionados. Finalmente, a través del uso de una técnica de reconocimiento de parámetros de control en la solicitud, se genera el flujo final con los servicios deseados, el cual es presentado a los usuarios finales y representa la salida del sistema.

## 5. PROTOTIPO Y EVALUACIÓN

La discusión de los retos y problemas en la Sección 2.2.3.2, permitió concluir que la recuperación en entornos convergentes es aún una tarea compleja, especialmente para usuarios finales (usuarios sin habilidades técnicas o de programación).

Para superar estas limitaciones, en los Capítulos 3 y 4, se propuso un enfoque que soporta la recuperación automática de servicios en entornos convergentes, con el fin de reducir la complejidad en la selección de servicios a usuarios finales, permitiéndoles especificar sus solicitudes a través de lenguaje natural. Para esto, se identifican criterios funcionales y no funcionales a partir, los cuales se utilizan en los procesos de emparejamiento y recomendación. Todo esto con el fin de brindar una mejor experiencia y mayor satisfacción al usuario final.

Este capítulo aborda la descripción detallada del desarrollo de un sistema software, que implementa la propuesta del proyecto. En la descripción se abarca la arquitectura definida para soportar la construcción del prototipo, los modelos de caso de uso del sistema, detalles de implementación del prototipo y el plan de pruebas del mismo, junto con los resultados obtenidos.

### 5.1. DESCRIPCIÓN DEL PROTOTIPO

El prototipo construido consiste en una aplicación desarrollada en lenguaje Java, implementada a partir de la adopción del Modelo de Construcción de Soluciones [80], como referencia metodológica para soportar el proceso de desarrollo. Las secciones siguientes, se centran en la aplicación de dos de los componentes más relevantes de ese modelo: Estructura para la Descripción del Sistema y el Modelo del Proceso de Desarrollo. El primero de ellos involucra la especificación de los elementos más importantes, para el entendimiento adecuado de la funcionalidad y el comportamiento esperado del sistema/solución (Modelo de Casos de Uso, Arquitectura de Referencia, Modelo de Diseño, Modelo de Despliegue y Plan de Pruebas); mientras que el modelo del proceso de desarrollo define cuatro fases de referencia, las cuales, ejecutadas en conjunto de forma iterativa e incremental, orientan el proceso de desarrollo de la solución propuesta.

#### 5.1.1. Modelo de Casos de Uso del Sistema

La definición de los casos de uso del sistema, parte de la identificación de los requisitos funcionales del mismo. A continuación se detallan los requisitos funcionales abstraídos del sistema implementado, los cuales son representados en la Figura 11.

- a. **Enriquecer términos semánticamente.** Con el conjunto de tokens o palabras simples obtenidas del proceso de tokenización de la solicitud y el conjunto de etiquetas descriptoras de servicio obtenidas a partir de un repositorio, el sistema debe enriquecer semánticamente estos términos a partir de dos fuentes de conocimiento (Wordnet y Flickr).
- b. **Generar una lista ordenada de servicios, comparando sintácticamente etiquetas de servicios con términos ingresados por los usuarios finales.** A partir de los términos procesados y analizados por el sistema (solicitud en lenguaje natural) y el enriquecimiento semántico del punto anterior, el sistema debe comparar a través de un método sintáctico, los términos de entrada con las etiquetas que describen los servicios y adicionalmente deberá organizarlos de mayor a menor nivel de similitud.
- c. **Determinar el flujo de control genérico.** El sistema debe determinar el flujo de control genérico, haciendo uso del conjunto de palabras de conjunción y orden obtenidas de la solicitud inicial.

- d. **Recomendar flujos de control genérico previamente almacenados.** Con el conjunto de flujos almacenados en un repositorio, el sistema debe reconocer cuál se asemeja más a la solicitud inicial, para posteriormente recomendar uno de ellos al usuario final.
- e. **Generar una lista ordenada de servicios de acuerdo a los valores de sus propiedades no funcionales.** El sistema debe evaluar las propiedades no funcionales de los servicios, asignando mayor peso a aquellas que han sido seleccionadas por el usuario, finalmente debe mostrar los servicios ordenados de acuerdo a los mejores valores de las propiedades encontradas en cada uno de ellos. El conjunto de servicios usados, están conformados por los servicios más importantes del primer ranking mostrado en el ítem II.
- f. **Identificar y clasificar palabras relevantes a partir de una solicitud en lenguaje natural.** Posterior al ingreso de la solicitud de un usuario, el sistema debe identificar las palabras que ingresa un usuario como también descartar aquellas que no sean relevantes

A partir del análisis de los requisitos funcionales del sistema, se definieron ocho casos de uso los cuales se describen a continuación:

**Caso de Uso No. 1: *Analizar Solicitud de usuario en lenguaje natural.***

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** Analizar semántica y sintácticamente una solicitud realizada en lenguaje natural por un usuario.

**Resumen:** Este caso de uso cumple los requisitos funcionales (f). Obtiene un conjunto de palabras sintácticamente clasificadas y semánticamente enriquecidas con el objetivo de capturar las palabras claves de la solicitud.

**Caso de Uso No. 2: *Enriquecer semánticamente.***

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** Enriquecer semánticamente las palabras ingresadas por los usuarios y las etiquetas que describen parámetros funcionales de los servicios.

**Resumen:** Este caso de uso soporta el requisito funcional (a). A partir del conjunto de palabras ingresadas por los usuarios y el conjunto de etiquetas almacenadas que representan las características funcionales de los servicios (interfaces de entrada/salida, elementos de comportamiento) y propiedades no funcionales genéricas, se hace un enriquecimiento semántico, a través de la fuente de conocimiento Wordnet y la fuente de conocimiento social Flickr de la cual es obtenido un conjunto de palabras relacionadas.

**Caso de Uso No. 3: *Generar ranking con Propiedades Funcionales***

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** Obtener el conjunto ordenado de servicios más relevantes de acuerdo a las palabras ingresadas por los usuarios.

**Resumen:** Este caso de uso obedece al requisito funcional (b), donde una vez obtenidas todas las palabras de entrada de la solicitud, se genera mediante un proceso de comparación sintáctica una lista con el conjunto de servicios que más se asemejan a los términos de entrada del usuario.

**Caso de Uso No. 4: *Generar ranking con Propiedades No Funcionales***

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** Obtener el conjunto ordenado de servicios más relevantes de acuerdo a los valores de propiedades no funcionales de cada servicio.

**Resumen:** Este caso de uso satisface el requisito funcional (e). El ranking es obtenido a partir de la evaluación de las propiedades no funcionales de los servicios y del conjunto de propiedades seleccionadas por los usuarios.

**Caso de Uso No. 5: *Generar Flujo de Control.***

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** Generar un flujo aproximado de control que relacione los servicios identificados o buscar a través del repositorio un flujo relacionado

**Resumen:** Este caso de uso cumple el requisito funcional (c), obtiene un flujo de control a partir de las palabras de control identificadas de la solicitud y el conjunto de servicios identificados como relevantes para el usuario.

**Caso de Uso No. 6: Recomendar Flujo de Control.**

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** A partir de la realización de un análisis de lenguaje natural, se realiza una recomendación de flujo de control que relacione las palabras de entrada con los flujos previamente almacenados

**Resumen:** Este caso de uso cumple el requisito funcional (d), obtiene un flujo de control a partir de la revisión de un repositorio de flujos.

**Caso de Uso No. 7: Recuperar Servicios**

**Iniciador:** Usuario (Agente Externo: Persona, Aplicación)

**Propósito:** Obtener el conjunto de servicios Web y Telco, mas adecuados de acuerdo a las propiedades funcionales y no funcionales de cada uno de ellos.

**Resumen:** Este caso de uso cumple los requisitos funcionales (b), (c) y (e). Obtiene el conjunto de servicios mas relevantes disponibles para un usuarios.

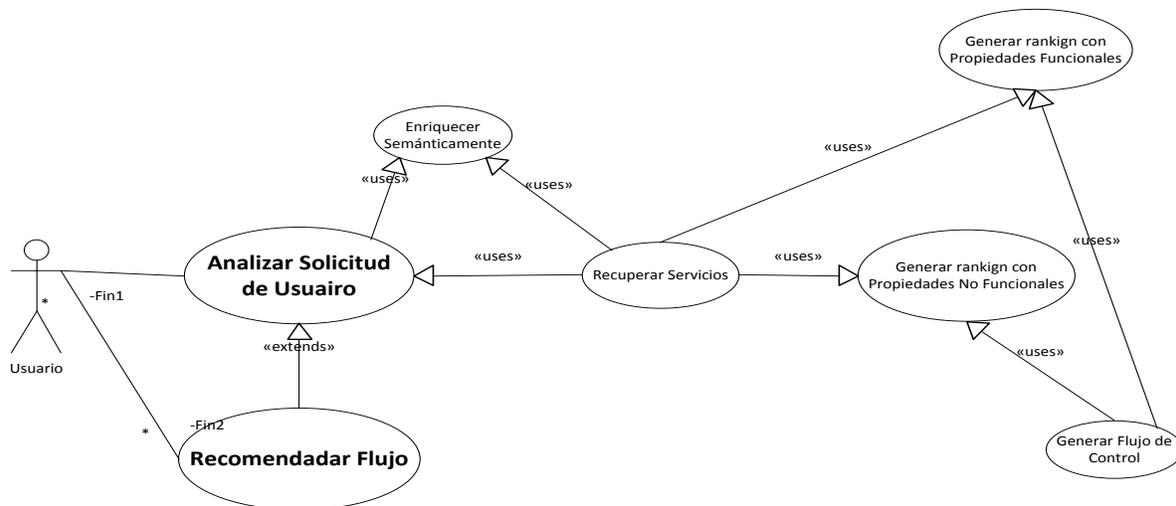


Figura 11. Diagrama de Casos de Uso del Sistema.

Adicional a los casos de uso definidos, es pertinente definir una arquitectura de referencia, que proporcione una visión general de los componentes del sistema. En la siguiente sección, se describe la arquitectura propuesta para el sistema implementado.

**5.1.2. Descripción de la Arquitectura de Referencia**

El prototipo desarrollado para evaluar el mecanismo de recuperación automática de servicios propuesto en este documento, adopta una arquitectura de aplicación en tres capas: la capa de Lógica de Presentación, la capa de Lógica de Negocio y una capa de Soporte. La Figura 12, presenta un diagrama de la arquitectura de referencia del sistema, en la cual se observa como la capa de Lógica de Presentación es implementada mediante una aplicación de escritorio y una aplicación móvil. Esta recibe la petición de un usuario representada a través de una solicitud realizada en lenguaje natural. Posteriormente, esta petición es

tramitada a la capa de Lógica de Negocio, donde principalmente se llevan a cabo siete procedimientos de gran importancia:

1. Analizador de Lenguaje Natural (Inmersas tareas de análisis de lenguaje natural): recibe información proveniente de la lógica de presentación.
2. Recomendador de etiquetas: interactúa directamente con el analizador de lenguaje y el repositorio de etiquetas de servicios, recomendando a los usuarios palabras que están directamente relacionados con los servicios disponibles en el repositorio y que han sido usados previamente por otros usuarios.
3. Enriquecedor semántico: recibe información de los procesos de análisis de lenguaje natural y del repositorio de servicios etiquetados, para llevar a cabo tareas de expansión de palabras de la solicitud del usuario.
4. Generador de Ranking: este proceso recibe información del analizador de lenguaje natural y la utiliza para la generación de listas que contienen un conjunto de servicios atómicos.
5. Generador de Flujo de Control: hace uso del analizador de lenguaje y del generador de ranking para obtener la salida del sistema.
6. Recomendador de Flujo de control: hace uso del repositorio de Flujos de control, con el fin de relacionar términos de la consulta del usuario con flujos previamente establecidos y de alguna manera mostrar este resultado como la salida del sistema
7. Ejecución de servicios: en el cual son ejecutados los diferentes servicios Web y de Telecomunicaciones y retorna si existe, la respuesta tras su ejecución.

El comportamiento definido en las capas superiores de la aplicación, se sustenta en un conjunto de APIs y proyectos de soporte. A continuación son descritos de manera breve:

- API Gate: soporta esencialmente los procesos de análisis de lenguaje natural.
- API Graphviz: soporta el proceso de generación del flujo de control.
- APIs de WordNet y Flickr: soportan el proceso de emparejamiento.
- Proyecto Jortho: el cual trata elementos de recomendación para la lógica de presentación.
- Proyecto Similarity: que tiene en cuenta el algoritmo de Lesk adaptado para la desambiguación semántica usada en el proceso de análisis de lenguaje natural
- API EasyWSDL: que permite leer, escribir, editar y crear WSDL 1.1, 2.0 WSDL y esquemas XML.
- API Apache CFX: que también ayuda a crear y desarrollar servicios utilizando las API de interfaz de programación JAX-WS y JAX-RS.

Adicionalmente, se establecen dos repositorios (Ver **Anexo B**) usados por la capa de la Lógica de Negocio, uno de ellos cuenta con etiquetas de propiedades funcionales y no funcionales de diferentes servicios disponibles, los cuales se utilizan en diversos procesos de esta misma capa. Por otra parte, se encuentra el repositorio con cada uno de los flujos de control resultantes a partir de las solicitudes de usuario, utilizado para realizar recomendaciones en futuras consultas por parte de usuarios finales.

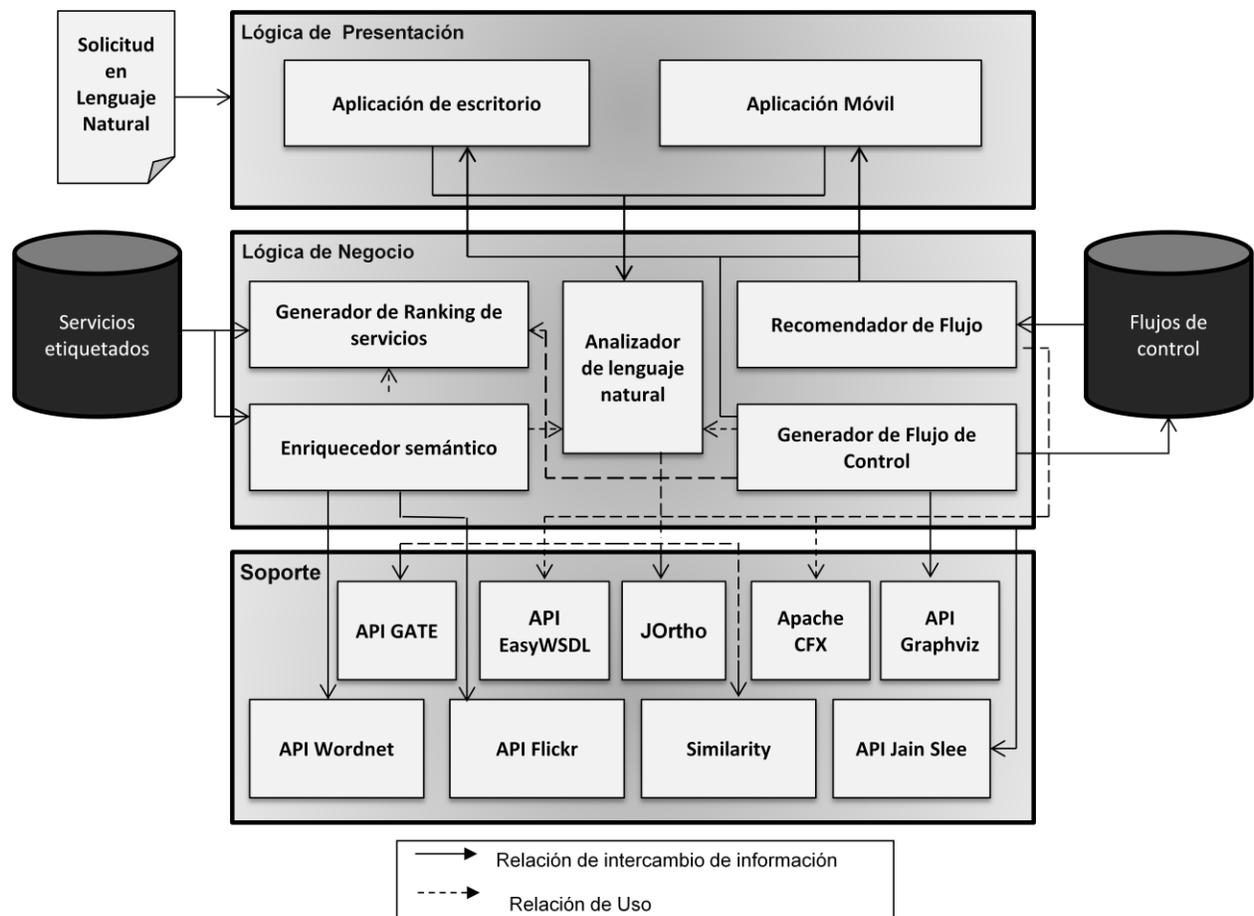


Figura 12. Arquitectura de Referencia del Sistema

A continuación, se aborda una descripción detallada de los componentes de la arquitectura de referencia del sistema.

### 5.1.2.1. SOPORTE

Esta capa inferior de la arquitectura está constituida por un conjunto de herramientas software, que hacen posible el desarrollo del proceso de recuperación de servicios. Las APIs que hacen parte de esta capa pueden encontrarse en el **Anexo D**.

### 5.1.2.2. Software de implementación de la capa de soporte

Como sistema operativo host fue elegido Windows 7 home Premium, el cual permite la ejecución de todas las APIs y proyectos mencionados anteriormente. Respecto al entorno de desarrollo, fue seleccionado Eclipse Helios como el IDE para la elaboración del prototipo. Adicionalmente se tuvieron en cuenta las siguientes consideraciones para su escogencia:

1. Es un IDE multiplataforma implementado y desarrollo en lenguaje Java.
2. Las APIs mencionadas Gate, Graphviz, Flickr y Wordnet, están disponibles para trabajarlas como librerías en dicho entorno.

A continuación se describen especificaciones técnicas de las herramientas usadas en el sistema implementado:

**Tabla 8. Especificaciones del sistema.**

Herramienta	Versión
JDK	1.6.0.20
Rhino SLEE	2.3.0.1
PostgreSQL	8.4

En cuanto al desarrollo de las interfaces de usuario y de salida del sistema se utilizó Java AWT para la aplicación de escritorio y del SDK de Android para la aplicación móvil.

### **5.1.2.3. Lógica de negocio**

En esta capa se implementan y articulan, todos los mecanismos necesarios para analizar las solicitudes de un usuario y obtener de la misma forma los elementos necesarios para la recuperación de servicios deseados. Este nivel de la arquitectura es compuesto por cinco unidades funcionales, las cuales interactúan con la capa de Lógica de Presentación y la capa de Soporte: Generación de flujo de control, Generación de Ranking de Servicios, Recomendador de Flujo de Control, Enriquecedor semántico y PLN.

#### ***Análisis de Lenguaje Natural.***

En este bloque se llevan a cabo todos los procesos necesarios para ejecutar el análisis del lenguaje natural. Inicialmente, una solicitud en lenguaje informal, escrita por un usuario, es introducida al sistema y esta, a través de la Lógica de presentación, pasa al macro-proceso de Análisis de lenguaje Natural. En este punto se utiliza principalmente el API de GATE y sus capacidades expuestas a través de ANNIE (*A Nearly New Information Extraction System*). En este proceso se ejecutan secuencialmente un conjunto de tareas: Tokenización, Filtrado de palabras, Etiquetado de palabras y Lematización, descritas en la Sección 3.1.2.

A su vez, aquellas que efectúan procesos más complejos, implicando un mayor consumo de recursos y análisis son: i). Clasificación de Nombres de Entidad (dedicada a la clasificación de palabras en diversas categorías). El objetivo de este módulo, es propiciar una clasificación de palabras en cuanto a las diferentes características de un servicio, como su comportamiento e interfaces, lo que conlleva además hacia una identificación más precisa de palabras que representan/describen a un servicio (Sección 3.1.2.5) ii). Desambiguación Semántica (tiene como fin realizar un proceso de resolución de significados de una palabra, cuando ésta posea diversos sentidos) mostrado en la Sección 3.1.2.5.

Este grupo de tareas retorna como salida a una lista de palabras relevantes respecto a la solicitud del usuario, las cuales a su vez, se encuentran ordenadas y clasificadas de acuerdo con el modelo de la solicitud del usuario propuesto en la sección 3.1 (Figura 5). De igual manera, esta salida se utiliza por otros procesos dentro de esta misma capa de Lógica de Negocio, corroborando su importancia dentro del sistema.

#### ***Enriquecedor semántico.***

Hace uso de las APIs WordNet y Flickr, quienes permiten encontrar y ampliar el número de términos asociados o relacionados a una palabra inicial ingresada por un usuario. Este tipo de enriquecimiento puede estar clasificado en dos tipos, según sea el grupo de elementos objetivos a enriquecer.

1. Términos de entrada: todas las solicitudes que atraviesan el proceso de análisis de lenguaje natural, pueden enriquecerse semánticamente a través de las dos APIs, cada vez que sean ingresadas por un usuario.

2. Términos descriptores de servicios. Adicionalmente, es pertinente ampliar el conjunto de datos almacenados que describen los servicios, ya que muchas veces el procesos de emparejamiento sintáctico no coincide exactamente con los elementos a evaluar, por lo cual un enriquecimiento, amplía no solo el número de etiquetas descriptoras, sino también la exactitud en el resultado de los servicios recuperados.

El proceso de enriquecimiento comienza a partir del uso de WordNet a través del API JAWS, con la cual se obtiene un total de tres palabras por cada una de las relaciones de: Sinónimos, Hiperónimos, “*Similar a*” y “*también ver*”.

Asimismo, haciendo uso del API de Flickr, que corresponde a una fuente de conocimiento social, se escoge el primer clúster recuperado con las primeras tres (3) etiquetas descritas, ya que al explorar los clúster con más de 3 etiquetas, se recuperan palabras que no poseen relación directa con el término inicial, similar a lo que sucede cuando se explora más de 1 clúster en particular (Ver Sección 4.1).

**Generador de Control de Flujo.** Constituye un módulo fundamental para la obtención final de la salida del sistema. Este bloque posee dos entradas que provienen del análisis de lenguaje natural y del generador de ranking. En primera instancia, este módulo emplea el API de Graphviz, con lo cual se establece una secuencia del flujo aproximado con bloques vacíos, a partir de la entrada de términos de orden. Conseguido este primer flujo, se procede con la entrada del *Generador de Ranking*, con el fin asociar cada uno de los rankings con los bloques generados del flujo mencionado. Una vez concluido este proceso, se establece la salida del sistema, la cual esta compuesta a su vez por una imagen que representa el flujo de control y también el orden de ejecución de los servicios recuperados. Adicionalmente, la salida se asocia directamente con el repositorio de flujos de control, donde es almacenada, con el propósito de utilizarla como recomendación (Ver Sección 4.4.2).

**Recomendador de Flujo y de términos.** Una de las características clave de mayor importancia en este tipo de proyectos, en los cuales el actor central es el usuario final, constituye precisamente la capacidad de interacción con el usuario y la respuesta del sistema. Esta última se debe ejecutar en momentos oportunos, con la particularidad de presentar un bajo consumo de recursos (bajos tiempos de ejecución). En primer lugar el proceso de recomendación de flujo, emplea un historial de flujos, obtenidos a partir de la satisfacción que usuarios previos han tenido y de la relación existente entre las palabras de entrada y las palabras que describen dichos flujos. Adicionalmente, se tiene la recomendación de términos, la cual está relacionada directamente con las palabras que un usuario ingresa en su solicitud, dado que, gracias a las capacidades que brindan el API AWT y Android con los sistemas de autocompletado, es posible sugerir términos a los usuarios que describen los servicios (Ver Sección 4.4.3).

**Generación del Ranking de servicios.** Como otro proceso de suma importancia dentro de la capa de Lógica de Negocio, se encuentra la generación del ranking de servicios. Este permite determinar el nivel de satisfacción de un usuario al ingresar una solicitud, basándose en el conjunto de servicios recuperados y ordenados por el sistema. Para soportar este proceso, se han desarrollado dos formas de ranking:

1. Generación del ranking Inicial: se genera a partir de un conjunto de etiquetas que describen las entradas/salidas de un servicio y su funcionalidad. El ranking se establece de acuerdo al mayor número de coincidencias encontradas entre el conjunto de palabras de entrada y el conjunto de palabras que describen un servicio. Consecuentemente, se establecerá una lista, priorizando aquellos servicios con mayor grado de similitud. Este proceso se llevo a cabo adaptando características específicas del proyecto WSColab [33] y adicionando procesos de enriquecimiento semántico, con el fin de mejorar la precisión del proceso de emparejamiento (Ver Sección 4.2).
2. Generación a partir de Propiedades No Funcionales: Este ranking se genera a partir de la evaluación y medición de los valores de propiedades no funcionales, asociadas a los servicios obtenidos en el primer

ranking. Este procedimiento tiene en cuenta el conjunto de propiedades no funcionales seleccionadas por los usuarios mediante las interfaces que componen la lógica de presentación. Al conjunto de propiedades seleccionado, le es asignado un mayor peso durante la generación del ranking. Esto se debe a que representa una prioridad mayor para los usuarios finales. En consecuencia, el ranking final es obtenido mediante un proceso matemático que considera, además de propiedades funcionales, los valores de las propiedades no funcionales de un servicio en particular, ofreciendo por tanto, servicios que contienen las mejores características y se ajustan a las preferencias del usuario (Ver Sección 4.3).

**Repositorio de servicios etiquetados.** De acuerdo al **Anexo B**, que especifica el diagrama de entidad-relación del repositorio de servicios etiquetados, este repositorio registra el conjunto de servicios disponibles del sistema, el cual tiene asociado:

1. Un conjunto de palabras dadas por el usuario que describen funcionalmente un servicio, son enriquecidas constantemente a través del proceso de recuperación de servicios o establecidas por el sistema inicialmente. Dentro de las tablas mencionadas, existe un atributo relevante, que representa el número de usuarios que usan un mismo elemento (palabra) como etiqueta de un servicio, el cual se utiliza dentro del proceso de emparejamiento sintáctico de servicios, en la generación de ranking inicial.
2. Un conjunto de palabras que describen propiedades no funcionales del servicio y sus valores respectivos, las cuales fueron tomadas de diversas fuentes en Internet, a través de repositorios de servicios Web disponibles como: Seekda, QWS Dataset y un conjunto de servicios de telecomunicaciones, los cuales fueron anotados manualmente por usuarios expertos. Este conjunto de propiedades no funcionales puede ser extendido y modificado, de acuerdo a las propiedades no funcionales en la medida se encuentren y utilicen distintos repositorios en la Web.

Cabe resaltar que el conjunto de servicios Web dispuestos en este repositorio, se obtuvo gracias a una herramienta desarrollada en el proyecto, la cual permitía la captura de un número amplio de servicios disponibles del servidor de Seekda y del QWS Dataset. En contraste, los servicios de telecomunicaciones fueron completamente desarrollados dentro del entorno Rhino Slee.

De esta manera, el repositorio de servicios, contiene la referencia de archivos que brindan un mayor soporte a la especificación de los servicios, para el caso de servicios Web, los archivos son sus WSDL correspondientes, usadas también para su consumo desde el prototipo, y para el caso de servicios de telecomunicaciones, son archivos de configuración creados en el entorno convergente de Rhino.

**Repositorio de Flujos de control.** Para el proceso de recomendación es importante almacenar el conjunto de términos asociado a un flujo de control en particular; para esto, el repositorio de flujo de control almacena: el grafo correspondiente al flujo que ha sido generado a partir de la solicitud del usuario final; el conjunto de palabras de la solicitud que relacionan dicho flujo y los servicios que fueron ejecutados.

#### **5.1.2.4 . Lógica de presentación**

Esta capa expone al usuario las funcionalidades ofrecidas por los componentes pertenecientes a la capa de Lógica de Negocio a través de una aplicación de escritorio o móvil, según sea el dispositivo que el usuario tenga a su disposición. En general estas aplicaciones disponen de un conjunto de vistas, que satisfacen los casos de uso definidos en la sección anterior, desde la perspectiva del usuario (ver **Anexo D**).

## **5.2. PRUEBAS DEL PROTOTIPO**

Una vez concluida la implementación del prototipo, es necesario someterlo a un proceso de evaluación para identificar y corregir deficiencias en la operación del sistema, con el fin de garantizar su óptimo desempeño. Las actividades de evaluación se enfocan en los dos componentes principales de la propuesta: el módulo de Procesamiento de Lenguaje Natural y el de Recuperación de Servicios. El objetivo general de esta

evaluación, consiste en determinar dos aspectos esenciales: i) rendimiento del sistema en términos de tiempo de ejecución y ii) validación de los módulos implementados comparándolos con mecanismos similares existentes. Esta comparación sigue la metodología de *Benchmarking*. Básicamente, un *Benchmarking* es el proceso encargado de comparar *benchmarks*. Un benchmark es el resultado de una evaluación hecha a un servicio, producto o proceso. Así, benchmarking es el proceso en el cual se puede comparar los resultados de evaluaciones hechas a distintos servicios, productos o procesos [138]. Cabe resaltar que el diagrama de clases usado en el desarrollo del prototipo se encuentra descrito en el **Anexo B**.

### 5.2.1. Módulo de Procesamiento de Lenguaje Natural

Con el fin de llevar a cabo las actividades de evaluación sobre el componente de procesamiento de lenguaje natural, fue necesario definir dos evaluaciones: la primera enfocada en evaluar la salida final del módulo para determinar la viabilidad de la propuesta; como entradas las solicitudes realizadas en lenguaje natural, a partir de la cual, obtiene términos que definen la descripción de un servicio a través de sus entradas/salidas, comportamiento, e incluso, términos que definen una lógica de flujo de los servicios (p.ej., “y”, “o”, etc.). La segunda evaluación consiste en seleccionar las tareas más importantes del proceso general de PLN implementadas y realizar las medidas de evaluación correspondientes, de acuerdo a criterios individuales, con el fin de evaluar la calidad de los resultados obtenidos por cada tarea.

#### 5.2.1.1. Evaluación del Módulo de Procesamiento de Lenguaje Natural

Para la evaluación general del módulo, se recolectaron un conjunto de 45 solicitudes en lenguaje natural por parte de usuarios de la Web. Para ello, se realizó una encuesta en la red Social de Facebook con la aplicación conocida como *Encuesta*<sup>27</sup>. La encuesta contenía una única pregunta de tipo abierta: “¿Si tuvieras en tu móvil una aplicación que te solucione diferentes problemas, permitiendo agilizar tus actividades diarias, qué le solicitarías? (formula la pregunta que le harías a la aplicación móvil... intenta imaginar una situación real.)”. Cada una de las solicitudes fue clasificada en una de tres categorías: alto, medio o bajo alcance. De acuerdo a su contenido, las solicitudes de bajo alcance representan solicitudes de servicio, sin significado en entornos convergentes, que no pueden ser implementados debido a su complejidad (p.ej., “dame esos cinco!!! jaja es broma”). Las de mediano alcance se refiere a solicitudes de servicio ambiguas, que pueden ser aplicadas al sistema, sin embargo contienen en su mayoría términos de difícil identificación en el contexto de servicios Web y de Telecomunicaciones (p.ej., “quiero un servicio que me haga soñar”). Finalmente, las solicitudes categorizadas como de alto alcance, corresponden a aquellas que contienen una información adecuada sobre los servicios, ya que se describen elementos y requisitos en términos de la funcionalidad del servicio y/o sus entradas/salidas y además, cumplen con una redacción coherente.

De la categoría alta y media descritas, se seleccionaron de manera aleatoria 8 solicitudes, las cuales fueron tomadas como la entrada del sistema. Para la evaluación, se consideraron dos aspectos importantes: por un lado, la clasificación de las palabras de la solicitud organizadas en grupos de Comportamiento (funcionalidad del servicio), Entrada/Salida (Interfaz del servicio) y Control (Flujo genérico de ejecución de los servicios); por otro lado, se consideró también el número de bloques de servicios, que generaba cada solicitud y palabras asociadas a cada uno de ellos (Tal como se puede observar en el ejemplo del Capítulo 4).

La evaluación general del módulo consistió en medir el nivel de acuerdo o desacuerdo en las tareas relacionadas con la clasificación de los términos ingresados en la solicitud y el número de bloques de servicios identificados, para lo cual participaron 10 personas expertas, con alto conocimiento en el tema de Recuperación y Composición de Servicios.

---

<sup>27</sup> Disponible en <http://www.facebook.com/mis.encuestas?ref=ts>

El formato de evaluación consistió de 8 cuestionarios (conjunto de preguntas), cada uno describiendo el procesamiento de una solicitud, mostrando la clasificación de los términos y la identificación del número de bloques de servicios con los términos de la solicitud asociados a cada bloque. Con los 8 cuestionarios se obtuvo un total de 77 preguntas de acuerdo/desacuerdo sobre las tareas del módulo y 26 preguntas de validación general del módulo (**Anexo E**). Para determinar el rendimiento del sistema, fue definido un conjunto de medidas que relacionan los resultados de los cuestionarios, las cuales se describen a continuación:

### **5.2.1.2. Medidas de Desempeño**

En este caso, las medidas de desempeño siguen un criterio binario, es decir, dada la salida del módulo de PLN, se mide el nivel de acuerdo o desacuerdo por parte de jueces con respecto a varios factores. Las medidas definidas para la evaluación del módulo de PLN son: acuerdo de clasificación de términos, acuerdo de estructura de solicitud formal y validación del módulo.

- **Acuerdo de clasificación de términos:** se refiere al nivel de acuerdo respecto a la capacidad que tiene el módulo para identificar y clasificar términos de la solicitud en lenguaje natural, con base en las categorías de comportamiento, entrada/salida y control.
- **Acuerdo de estructura de solicitud formal:** se refiere al nivel de acuerdo respecto a la capacidad que tiene el módulo para generar una estructura que indique de manera organizada la cantidad de bloques de servicios requeridos para cumplir la solicitud del usuario; y a su vez, se refiere a los términos que deben usarse para la búsqueda de cada uno de dichos bloques.
- **Validación del módulo:** indica la calidad de los resultados del módulo de PLN de acuerdo a los resultados obtenidos en las 2 medidas anteriores, pudiendo ser excelente, bueno, regular ó malo.

### **5.2.1.3. Evaluación de las Tareas de Procesamiento de Lenguaje Natural**

De las 6 tareas de PLN (Sección 3.1.2), aquellas que fueron seleccionadas para su evaluación dada su importancia, funcionalidad y aporte para el objetivo final del módulo (generación de términos de descripción de servicios y su lógica genérica de ejecución), son: etiquetado de palabras, reconocimiento de nombres de entidad y desambiguador semántico. Los formatos usados para la evaluación de estas tareas están disponibles en el **Anexo F**.

La evaluación de estas tareas resulta uno de los ejes importantes del artículo de investigación “*User-Driven Automatic Resource Retrieval Based on Natural Language Request*” [75], creado durante el trabajo de grado. En el paper, se describe la propuesta de recuperación de servicios en entornos convergentes orientado al usuario, centrándose en el mecanismo de procesamiento de lenguaje natural (descripción de los algoritmos, estructura de la solicitud de un usuario, estructura de un servicio, entre otros aspectos). El objetivo del artículo además de validar parte del trabajo de la monografía, fue presentar un descripción detallada de la evaluación experimental de las tareas de PLN, mostrando resultados que permiten ver un buen rendimiento en términos generales, considerando la complejidad de cada tarea por separado, además fue posible detectar aspectos por mejorar con el fin de incrementar la precisión del sistema para trabajos futuros (**Anexo E**).

### **5.2.2. Módulo de Recuperación de Servicios**

La evaluación en el módulo de recuperación de servicios se concentró en medir el desempeño tanto de la recuperación inicial de los servicios a partir de los términos recuperados de la solicitud, como del mecanismo de refinamiento del ranking con los valores numéricos de las propiedades no funcionales de los servicios del ranking inicial.

### 5.2.2.1. Metodología de Evaluación del Sistema de Recuperación de Servicios

Con el fin de llevar a cabo las actividades de evaluación, fue necesaria la creación de un banco de servicios que tuviera descripciones de su funcionalidad e interfaz. El banco representa un total de 185 servicios obtenidos de: Seekda (139 servicios), el repositorio público The QWS Dataset (23 servicios) [12] y servicios desplegados en un entorno de telecomunicaciones (6 servicios); este grupo de servicios alimentan el mecanismo de emparejamiento implementado. Los servicios fueron descritos usando los términos por defecto que algunos poseían y enriquecidos con anotaciones manuales realizadas por expertos (Estudiantes en proyecto de grado miembros el Grupo de Ingeniería Telemática), junto al enriquecimiento inicial de términos explicado en el capítulo 4. De esta manera, se tuvo un total de 846 términos de comportamiento, 3.476 términos de entrada/salida, 3.698 términos de comportamiento enriquecidos y 3.132 términos de entrada/salida enriquecidos, para un total de 11.152 términos que describen los 185 servicios. Estos servicios, junto a su descripción, fueron almacenados en una base de datos relacional descrita en el **Anexo C**.

Para la evaluación, se recogió un total de 28 términos generados por el módulo de PLN a partir de las 45 solicitudes en lenguaje natural, obtenidas de la encuesta realizada para la evaluación del módulo de PLN. Los 28 términos fueron distribuidos en 5 conjuntos y cada conjunto correspondió a la entrada del módulo de recuperación para efectuar el emparejamiento con los términos que describen los servicios y generar el ranking de servicios. La distribución de los términos en los conjuntos, se levo a cabo considerando características genéricas de la funcionalidad de los servicios así: mensajería, ubicación, comunicación, búsqueda y entretenimiento. De esta manera, los términos del mismo conjunto tenían relaciones fuertes que permitían la búsqueda de servicios relacionados con todos los términos. Adicionalmente, se consideró que los mismos términos pertenecieran a las categorías de comportamiento y de interfaz, para poder realizar comparaciones posteriores con el emparejador base WSColab, el cual recibe como entrada la categoría, comportamiento y salida por separado.

Para la obtención del *benchmark* de referencia, se pidió a los evaluadores expertos asignar un nivel de relevancia (que tan importante es el servicio, en un rango de valores de 0 a 100) a cada servicio con respecto a cada uno de los conjuntos de términos. Para ello, se diseñó un formato de evaluación que contenía los 185 servicios con las etiquetas descriptoras correspondientes y los 5 conjuntos de términos. De esta manera, se obtuvieron 925 comparaciones por evaluación (5x185), donde participaron 5 jueces expertos (Estudiantes en proyecto de grado y de maestría en el Grupo de Ingeniería Telemática). El promedio de los niveles de relevancia dada por los jueces, para cada conjunto fue considerado el *benchmark* de referencia, con el cual sería posible comparar el ranking de recuperación, resultado del enfoque propuesto, el cual también organiza los servicios de acuerdo a su valor de relevancia con respecto a los términos.

### 5.2.2.2. Metodología de Evaluación del Sistema de Recuperación basado en Propiedades No Funcionales

Para realizar la evaluación de este mecanismo, se creó un banco de propiedades no funcionales para los servicios obtenidos, descritos anteriormente. Las fuentes de los servicios, Seekda y The QWS Dataset, proporcionan valores numéricos en términos de porcentajes o unidades de medida (P.ej. milisegundos) de propiedades no funcionales de los servicios. En el caso de Seekda, se ofrecen las propiedades *disponibilidad* y *documentación*; en el caso de The QWS Dataset, se ofrecen las propiedades *tiempo de respuesta*, *disponibilidad*, *rendimiento*, *tasa de éxito*, *confiabilidad*, *conformidad*, *mejores prácticas*, *latencia* y *documentación*.

Para los servicios desplegados en el entorno de telecomunicaciones, se consideraron, además de las propiedades que deben cumplir las capacidades de telecomunicaciones, las propiedades no funcionales del entorno de telecomunicaciones, proporcionadas en la documentación [145], [146] obteniendo los valores de *rendimiento*, *tiempo de respuesta*, *documentación*, *mejores prácticas*, *disponibilidad*, *confiabilidad* y

*latencia*. Con este banco se obtuvo un total de 9 propiedades no funcionales diferentes y un total de 561 valores de las propiedades para los 185 servicios. Los nombres y valores de las propiedades no funcionales de los servicios fueron almacenados en una base de datos relacional descrita en el **Anexo C**.

Para la entrada del mecanismo, se crearon 3 solicitudes diferentes que consistían en posibles combinaciones de preferencias no funcionales. Además, se configuró el sistema para que únicamente considere las propiedades no funcionales de los servicios, a la hora de organizarlos de mayor a menor relevancia.

Para la obtención del *benchmark* de referencia, se pidió a los evaluadores expertos asignar un nivel de relevancia (que tan importante es el servicio, en un rango de valores de 0 a 100) a cada servicio con respecto a cada una de las solicitudes. El formato de evaluación para ello, contenía los 185 servicios con los valores de las propiedades no funcionales correspondientes. Por cada evaluación realizada, se crearon 555 comparaciones (3x185), realizadas por 5 jueces expertos (Estudiantes en proyecto de grado y de maestría del Grupo de Ingeniería Telemática). El *benchmark* de referencia se obtuvo con el promedio de los niveles de relevancia asignados para cada solicitud por los jueces, con el cual también fue posible generar el ranking de servicios de acuerdo a la relevancia de cada uno, con respecto a las propiedades no funcionales de la solicitud.

### **Métricas de Evaluación**

Por lo general, el proceso de emparejamiento (*matchmaking*) en el contexto de la recuperación de servicios, compara una solicitud (Lenguajes formales) de servicio con el conjunto de servicios disponibles y retorna una lista de servicios ordenados de acuerdo con la relevancia respecto a la solicitud. Las evaluaciones de estos sistemas, normalmente siguen el enfoque tomado en la evaluación de sistemas de recuperación de información. En estos sistemas, existe una variedad de métricas de evaluación con varias características. Sin embargo, en tales métricas no se ha tenido en cuenta la influencia de diferentes factores en la estabilidad y significado de los resultados (Por ejemplo considerar la importancia del orden en el cual se recuperan los resultados ó el nivel de prioridad que tienen los primeros ítems recuperados respecto al resto). Es por esa razón, que en el proyecto se tuvieron en cuenta las recomendaciones y medidas proporcionadas por el trabajo en [127], que permitieron tomar decisiones apropiadas sobre las métricas de evaluación, las cuales se describen a continuación.

Inicialmente, las métricas deben permitir ser comparadas sobre diferentes solicitudes de servicios [127]; en segundo lugar, las métricas deberían permitir configurar el grado de preferencia de ítems de mayor relevancia, con respecto a los de baja relevancia [127]. Finalmente, teniendo en cuenta el hecho de que en un ranking de servicios, por lo general se seleccionan los primeros ítems, las medidas deberían concentrarse en la calidad de los primeros ítems del ranking [127]. Por lo tanto, antes de describir las medidas que se usaron en esta fase de evaluación, es necesario introducir los términos usados para su descripción.

**Ranking:** un ranking  $r$  de un conjunto de servicios  $S$ , es una secuencia de elementos ordenados de  $S$  según su valor de relevancia respecto a una consulta de servicios determinada. Un ranking con todos los servicios disponibles, es llamado *ranking completo* [127].

**Ganancia:** La ganancia (mayor ó igual a 0) de un servicio  $s$  con respecto a una solicitud  $q$  denota la relevancia de  $s$  para  $q$ ; la función de ganancia es la función que asigna a cada servicio  $s$  de un ranking una ganancia con respecto a la solicitud  $q$  [127]. Se habla de Ganancia Ideal, cuando es la ganancia proporcionada por el benchmark de referencia.

**Ranking Ideal:** Un ranking es ideal si y solo si, lista los servicios en orden decreciente según la relevancia con respecto a una solicitud [127].

**Medida de Efectividad de Recuperación:** Es una función que asigna a un ranking  $r$  un valor entre 0 y 1 con respecto a la función de ganancia para dicho ranking [127].

Las métricas usadas, además de las bien conocidas *Recall* y *Precision*, corresponden a las medidas mejoradas, introducidas en [127] y son: *Averaged Normalized Cumulated Gain* (medida mejorada de Average Weighted Precision (AWP) [127]), *Averaged Normalized Discounted Cumulated Gain* (medida mejorada de Average Weighted Discounted Precision (AWDP) [127]) y GenAveP' (medida mejorada de Generalization Average Precision (GenAveP) [127]). A continuación se describe cada una de las medidas usadas.

- **Precision ( $p$ ):** proporción de los servicios relevantes recuperados por el emparejador a partir de la solicitud.
- **Recall ( $r$ ):** proporción de todos los servicios relevantes contenidos en los servicios recuperados por el emparejador a partir de la solicitud.
- **Overall ( $o$ ):** relaciona las medidas de *precision* y *recall*, y se emplea para determinar la calidad del proceso de recuperación de servicios.
- **Fixed Generalization Average Precision (GenAveP')**: definida en [147], es una generalización de la precisión promedio (Average Precision) [127], que captura el rendimiento del emparejador en una única medida, considerando las métricas de *recall* y *precisión*. GenAveP es la medida en que se penaliza la recuperación de los últimos ítems y se realiza por tanto, el promedio sobre todos los ítems sean relevantes o no.
- **Averaged Normalized Cumulated Gain (ANCG):** es una extensión de la precisión promedio [127], propuesta en [148], basada en la acumulación de la ganancia de los primeros  $i$ -servicios del ranking [127]. Los problemas que tiene la medida tradicional, es que el orden de los servicios en el ranking importan más que su valor de relevancia, además no permite configurar la medida en que se penaliza la recuperación de los ítems relevantes en las últimas posiciones en el ranking, como la medida anterior; por ello, según [127], se realiza el promedio sobre todos los ítems.
- **Averaged Normalized Discounted Cumulated Gain (ANDCG):** es una versión simple de la medida anterior, que usa un factor de descuento para penalizar la recuperación de los servicios relevantes del ranking en las últimas posiciones [127]. Esta medida, elige una función apropiada de descuento que permite configurar la medida en la cual la recuperación de servicios relevantes en últimas posiciones, es penalizado.

Estas medidas operan sobre diferentes conjuntos y ecuaciones: El conjunto de verdaderos positivos SP (servicios recuperados relevantes según el benchmark de referencia), el conjunto de falsos positivos FP (servicios recuperados no relevantes según el benchmark de referencia), el conjunto de falsos negativos FN (servicios que no son recuperados pese a que son relevantes de acuerdo al benchmark de referencia [127]), ganancia acumulada CG (suma las ganancias recibidas por los primeros  $i$ -servicios del ranking del emparejador [127]), ganancia acumulada ideal ICG (suma las ganancias recibidas por los primeros  $i$ -servicios del ranking del *benchmark* de referencia [165]), ganancia acumulada descontada DCG (suma las ganancias del servicio en el ranking, recibidas por los primeros  $i$ -servicios del ranking del emparejador, normalizadas por la raíz cuadrada de la posición  $i$  [127]) y ganancia acumulada descontada ideal IDCG (suma las ganancias del servicio en el ranking del benchmark de referencia, recibidas por los primeros  $i$ -servicios del ranking del emparejador, normalizadas por la raíz cuadrada de la posición  $i$  [127]).

Una vez aclarados los conjuntos sobre los cuales actúan las métricas descritas, a continuación se presentan las definiciones formales adaptas al contexto del presente trabajo de investigación.

**Precision ( $p$ ):** se refiere a la proporción del número de servicios recuperados relevantes respecto al número de servicios recuperados por el emparejador.

$$p = \frac{SP}{SP + FP} \quad (33)$$

**Recall (r):** corresponde a la proporción del número de servicios recuperados relevantes respecto al número total de servicios relevantes.

$$r = \frac{SP}{SP + FN} \quad (34)$$

**Overall (o):** relaciona las medidas de *precision* y *recall*, matemáticamente se calcula de la siguiente manera.

$$o = r * \left(2 - \frac{1}{p}\right) \quad (26)$$

**Fixed Generalization Average Precision (GenAveP')**: mide el desempeño en términos de la ganancia acumulada respecto a la ganancia acumulada de los servicios relevantes para la solicitud.

$$GenAvep' = \frac{\sum_{i=1}^{SP+FP} \frac{CG(i)}{i}}{\sum_{i=1}^{SP+FN} \frac{ICG(i)}{i}} \quad (36)$$

**Averaged Normalized Cumulated Gain (ANCG):** determina la proporción de ganancias acumuladas del ranking de servicios recuperados respecto a la proporción de ganancias del ranking del benchmark de referencia.

$$ANCG = \frac{1}{SP + FN} \sum_{i=1}^{SP+FP} \frac{CG(i)}{ICG(i)} \quad (37)$$

**Averaged Normalized Discounted Cumulated Gain (ANDCG):** mide el desempeño en términos de la ganancia acumulada normalizada con la raíz cuadrada de la posición *i* en el ranking respecto a la ganancia acumulada de los servicios relevantes para la solicitud, igualmente normalizados con raíz cuadrada de la posición *i*.

$$ANDCG = \frac{1}{SP + FN} \sum_{i=1}^{SP+FP} \frac{DCG(i)}{IDCG(i)} \quad (38)$$

La siguiente sección detalla el plan de pruebas dispuesto para el sistema implementado, junto con los resultados obtenidos a partir de su ejecución.

### 5.2.3. Plan de Pruebas y Resultados Obtenidos

Todas las pruebas a las cuales se hace referencia a continuación, se llevaron a cabo sobre el prototipo generado, al concluir la segunda iteración del proceso de desarrollo de la solución propuesta. De esta manera, se realizó la evaluación del desempeño de las funcionalidades implementadas en la Capa de Lógica de Negocio de la Arquitectura descrita en secciones previas.

### 5.2.3.1. Plan de Pruebas

A continuación se describen las pruebas de calidad y eficiencia ejecutadas sobre el prototipo desarrollado:

Tabla 9. Plan de Pruebas

PLAN DE PRUEBAS	
<b>Recuperación de términos relevantes de la solicitud en lenguaje natural</b>	<b>Prueba de Rendimiento RT1:</b> permite medir el tiempo que le toma al sistema llevar a cabo la recuperación de los términos de servicios y flujo de control, de la solicitud en lenguaje natural, agregando nuevos términos con algunas tareas.
	<b>Prueba de Calidad RC1:</b> permite evaluar la calidad de los resultados de los términos obtenidos por el sistema.
<b>Desambiguación de Sentidos de Palabras</b>	<b>Prueba de Calidad DC1:</b> permite evaluar la calidad del sistema para desambiguar una palabra correctamente y asignar el sentido correcto.
<b>Reconocimiento de Nombres de Entidad</b>	<b>Prueba de Calidad NC1:</b> permite evaluar la calidad del sistema para anotar términos como entidades correctamente.
<b>Etiquetado de Palabras</b>	<b>Prueba de Calidad EC1:</b> evalúa la calidad del sistema para anotar las palabras según su categoría gramatical de manera correcta.
<b>Recuperación y Refinamiento de Servicios</b>	<b>Prueba de Rendimiento RR1:</b> permite determinar el tiempo que toma el sistema en recuperar los servicios más relevantes para satisfacer los criterios funcionales y no funcionales.
	<b>Prueba de Calidad SC1:</b> permite evaluar la calidad de los resultados de similitud obtenidos por el sistema para la recuperación de servicios, al comparar una consulta con términos provenientes de solicitudes en lenguaje natural con los servicios disponibles, considerando criterios funcionales y no funcionales.

Los resultados de las pruebas de rendimiento, están condicionados por las prestaciones del equipo donde se ejecuta el software del prototipo. En este caso, el equipo empleado para desplegar el plan de pruebas descrito cuenta con las siguientes especificaciones técnicas:

Tabla 10. Especificaciones Técnicas del Equipo empleado para las Pruebas del Prototipo

<b>Microprocesador</b>	Intel Pentium SU4100 @ 1.30GHZ 1 procesador, 2 núcleos. Intel Core 2 Duo de 2.30GHz Genuine Intel(R) CPU U4100 de 1.30GHz Intel Pentium SU4100 @ 1.30 GHz 1 processor, 2 cores GenuineIntel Family 6 Model 23 Stepping 10
<b>Caché del microprocesador</b>	2 MB de caché de nivel 2
<b>Memoria</b>	4000 MB
<b>Disco duro</b>	250GB (5400 RPM)
<b>Sistema Operativo</b>	Windows 7 Home Premium
<b>Red</b>	Internet de Banda Ancha (ADSL) de 1 Mbps (Transmisión de 125 KB/s)

### 5.2.3.2. Resultados

Con la ejecución del plan de pruebas descrito, se obtuvo un conjunto de resultados, los cuales se representan gráficamente en esta sección, con el fin de facilitar su análisis y determinar, por tanto, el desempeño general del sistema desarrollado.

- **Recuperación de términos relevantes de la solicitud en lenguaje natural**

**Prueba de Rendimiento RT1:** Con la ejecución de esta prueba, se determinó el tiempo que toma el sistema en llevar a cabo la recuperación de los términos de servicios y flujo de control, a partir de solicitudes en lenguaje natural; a la vez que realiza tareas de expansión de los términos:

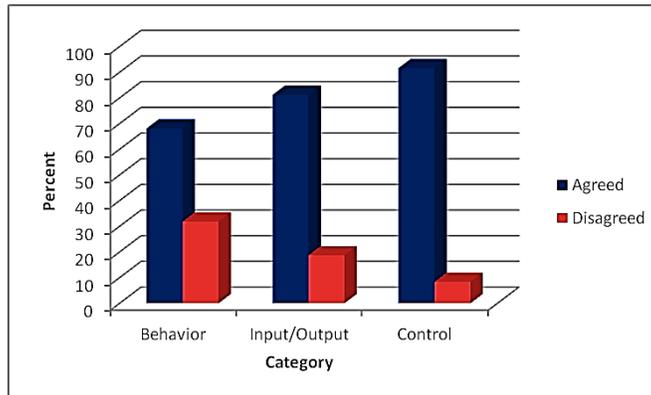


Figura 13. Gráfica de Rendimiento - Recuperación de términos relevantes de la solicitud en lenguaje natural.

El comportamiento de la curva en la Figura 13, muestra una relación exponencial entre el número de palabras de la solicitud y el tiempo que toma el sistema en la ejecución de esta actividad. Por otra parte, se considera que el tiempo empleado en esta operación es aceptable (hasta de 1 segundo para la obtención de términos con solicitudes menores ó iguales a 20 palabras, ya que la mayoría de solicitudes de usuarios se encuentran en este rango). La explicación de la tendencia exponencial de la figura se debe a la cantidad de tareas de PLN involucradas; algunas tareas complejas como la desambiguación con fuentes léxicas o mecanismos de patrones de reglas como el reconocimiento de entidades, las cuales desmejoran el desempeño entre mayor sea la cantidad de palabras que deben procesar. La tarea de PLN que requiere mayor tiempo de procesamiento, según sea el número de palabras usadas en la solicitud, es la de desambiguación, ya que el algoritmo de *Lesk* usado para esta tarea, a pesar de ser uno de los más consistentes, según se explicó en la Sección 3, realiza comparaciones de sentidos de cada palabra ambigua con los sentidos de las demás palabras de la solicitud; razón por la cual, entre mayor sea el número de palabras en la solicitud, mayor será el número de comparaciones de sentidos de palabras, incrementando el tiempo considerablemente.

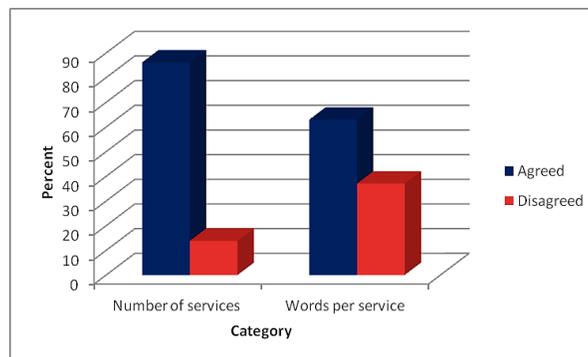
**Prueba de Calidad RC1:** Tras la ejecución de esta prueba, fue posible determinar la calidad de los términos obtenidos para una solicitud en lenguaje natural, en cuanto a las posibles descripciones de los servicios o el flujo lógico general en que se deberían recuperar. En la figura 14, se muestra el grado de satisfacción con respecto a la clasificación y selección de términos provenientes de la solicitud en lenguaje natural. El porcentaje más alto de consenso fue en la selección de términos del flujo de control con un valor de 91,6% y el más bajo se presentó en los términos de comportamiento, con un valor de 68,1%. En la categoría de entrada/salida, se obtuvo un porcentaje de acuerdo de 81,2%. Con relación a resultados obtenidos, el desempeño del sistema en cuanto a la identificación de términos de las diferentes categorías de solicitudes en lenguaje natural es en general adecuado. Este resultado puede mejorarse, si se trabaja en

la identificación de la categoría de comportamiento, con la definición de un conjunto ampliado de términos y reglas adicionales que describan este parámetro en la solicitud.



**Figura 14. Acuerdo de clasificación de términos.**

En la Figura 15, se muestra el nivel porcentual de “*acuerdo de estructura de la solicitud formal*”, es decir, la satisfacción con respecto al número de servicios identificados y los términos detectados para la búsqueda de cada servicio. El número de servicios identificados acordes a las solicitudes, mostró un fuerte nivel de consenso con el 86,1%, mientras que los términos asociados a cada servicio identificado, mostraron una satisfacción del 62,9%. Se puede decir que la estructura de la solicitud formal es considerada de buena calidad, aunque los términos usados para la recuperación de cada servicio pueden ser mejorados con nuevos mecanismos de filtrado y expansión de términos.



**Figura 15. Acuerdo de estructura de solicitud formal.**

La validación del modulo, que indica el desempeño general, es mostrado en la Figura 16, lo cual evidencia que el sistema es considerado bueno con porcentajes de 41,6% y 59,7% para las identificación de servicios y clasificación de términos respectivamente, mientras que solo una minoría lo considera malo con porcentajes por debajo del 3% para las dos actividades.

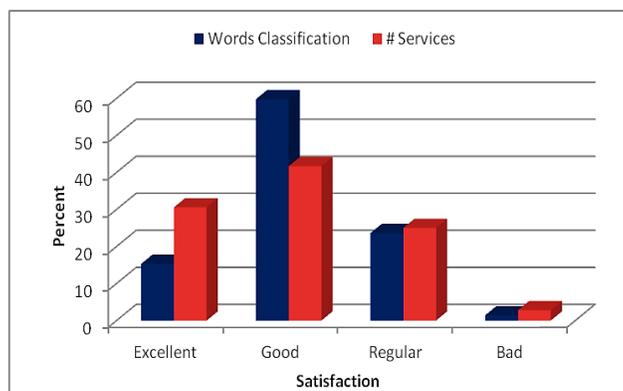


Figura 16. Validación de modulo.

- **Desambiguación de Sentidos de Palabras**

**Prueba de Calidad DC1:** Con la ejecución de esta prueba, se determina la calidad del sistema para desambiguar una palabra correctamente, proveniente de la solicitud en lenguaje natural. En la Figura 17, se presentan los resultados generales de las medidas de *precisión* y *recall*, evaluadas sobre diferentes palabras ambiguas. La Figura 17, reporta resultados por debajo del 50% tanto en *precisión* como en *recall*; esto se debe principalmente a que el contexto de la palabra a desambiguar poseía palabras comunes para los diferentes sentidos probables de la palabra, lo que incrementaba la posibilidad de error del algoritmo de Lesk. Por otro lado, también muestra resultados que superaron los sistemas participantes de Senseval-2<sup>28</sup>, cuyo mejor representante obtuvo un valor de 51,2% para *precisión* y *recall*. El desempeño del sistema se ve fuertemente influenciado por las palabras usadas en el contexto de la palabra ambigua, se puede decir que el desempeño no resulta tan adecuado cuando se usan palabras comunes; sin embargo, en la solicitud de los usuarios generalmente los términos, tienen una relación con los servicios, como el caso del contexto de la palabra *Actíivate* en la evaluación, por lo que los valores de *precisión* y *recall* para dichos casos es adecuado.

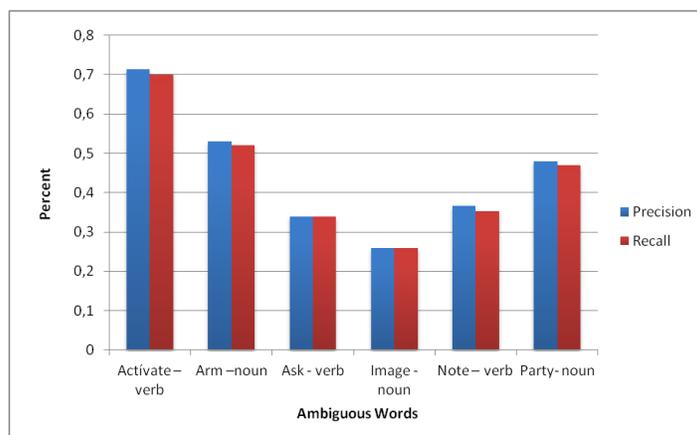
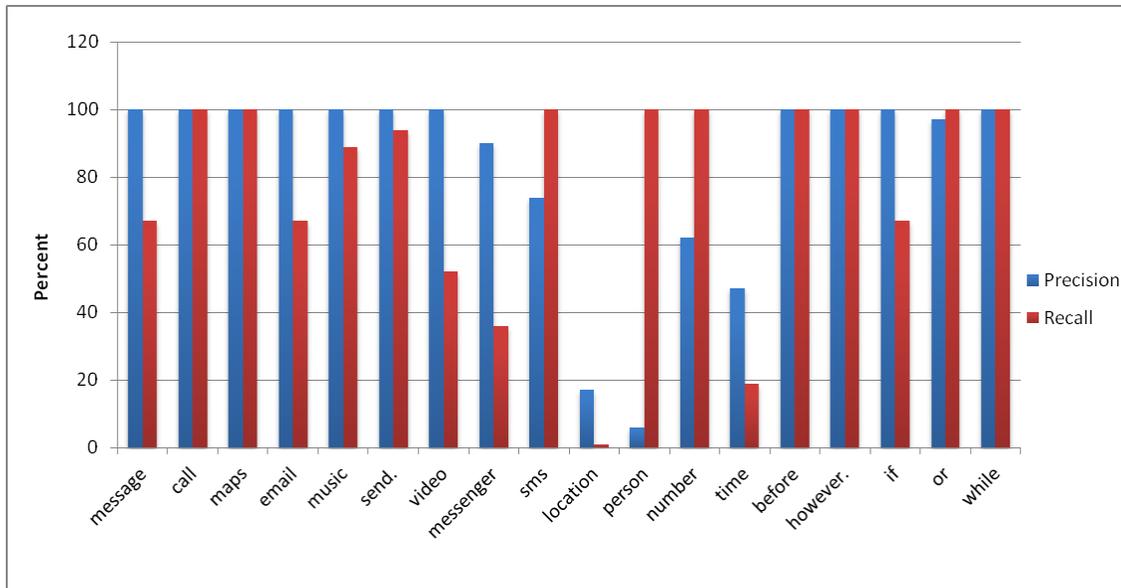


Figura 17. Medida de Desempeño: Precisión y Recall para la tarea de Desambiguación

- **Reconocimiento de Nombres de Entidad**

**Prueba de Calidad NC1:** En esta prueba, se evalúa el desempeño del sistema para clasificar entidades a partir de términos recuperados de la solicitud del usuario.

<sup>28</sup> Disponible en: <http://www.senseval.org/>



**Figura 18. Medida de Desempeño: Precisión y Recall para la tarea de Reconocimiento de Nombres de Entidad**

En la Figura 18, se presentan los resultados generales de las medidas de *precision* y *recall*, evaluadas sobre diferentes entidades definidas en el prototipo. La Figura 18, muestra resultados de *precisión* y *recall* por encima del 50%, lo cual es adecuado considerando que superan o igualan las medidas de *precisión* y *recall* de GATE y otros clasificadores con entidades definidas por defecto, presentadas en [142]. Sin embargo, las medidas de categorías como *location*, son consideradas como malos resultados, ya que el número de entidades realizadas por el sistema y las entidades anotadas correctamente fueron bajas comparadas con el número de entidades dados por el *benchmark* de referencia. Las diferencias en estos valores se deben a la falta de entidades de *location* en las listas definidas con *Gazetter*, que coincidieran con aquellas obtenidas por el corpus de referencia, también en el caso específico de esta entidad, la regla que condiciona la anotación de un término como tipo *location*, consideraba la palabra después de los términos “*in*” y “*at*”; lo cual fue demasiado genérica para distinguir una entidad real de un falso positivo.

- **Etiquetado de Palabras**

**Prueba de Calidad EC1:** en esta prueba, se evalúa el desempeño del sistema para determinar la categoría gramatical de las palabras ingresadas por el usuario en lenguaje natural. En la Figura 19, se presentan los resultados generales de las medidas de *precision* y *recall*, evaluadas sobre diferentes corpus de palabras. De la Figura 19, se puede decir que las medidas de *recall* y *precision* presentan un buen rendimiento con resultados por encima del 80%, demostrando que la mayoría de etiquetas de palabras en diferentes categorías fueron asignadas correctamente.

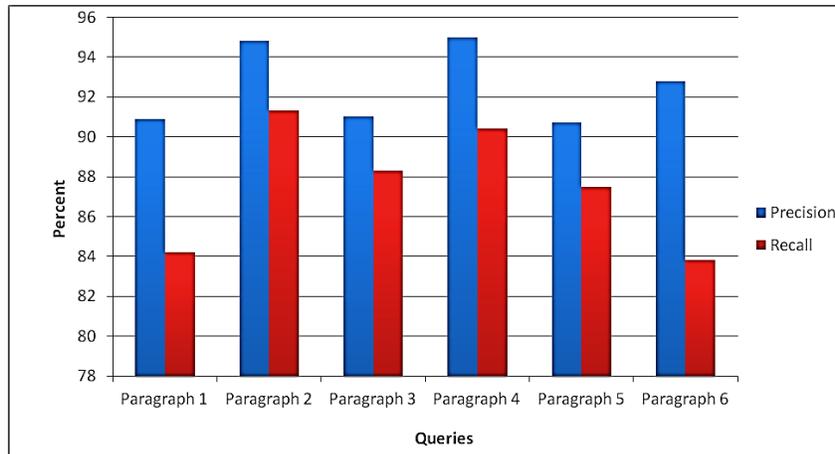


Figura 19. Medida de Desempeño: *Precision* y *Recall* para la tarea de Etiquetado de Palabras

- **Recuperación y Refinamiento de Servicios**

### Métricas Tradicionales de Evaluación

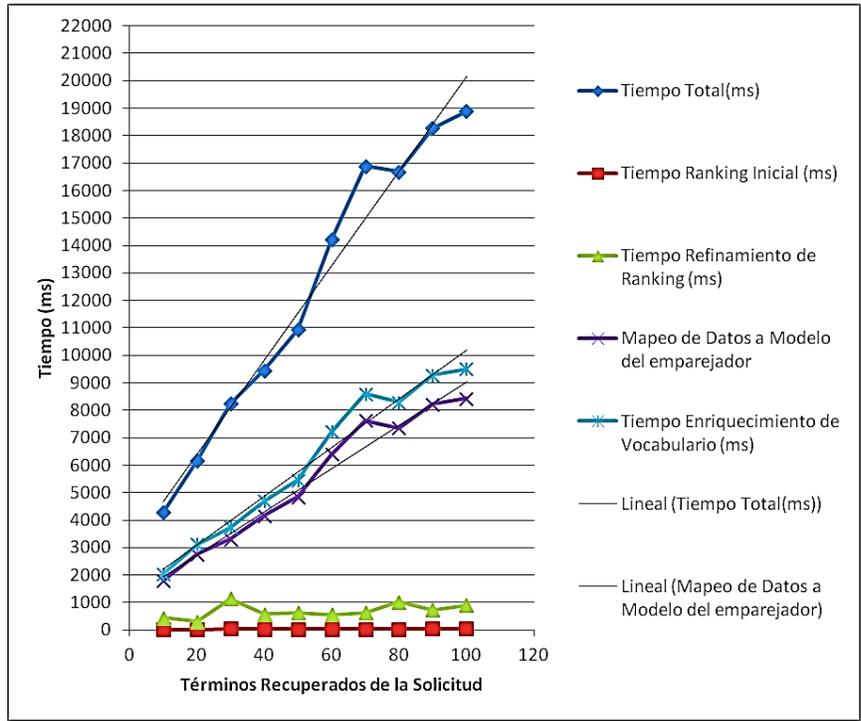
**Prueba de Rendimiento RR1:** Con la ejecución de esta prueba, se determinó el tiempo que toma el sistema en llevar a cabo la recuperación de los servicios considerando los criterios funcionales y los no funcionales disponibles en el prototipo. Este procedimiento se ejecuta desde el momento en que se recuperan los términos de la solicitud en lenguaje natural, considerando así varias fases: enriquecimiento de vocabulario, mapeo de los datos a un modelo que permite realizar la comparación en el emparejador, comparación de términos y servicios para la generación de un ranking inicial (Tiempo Ranking Inicial en la Figuras 20 y 21) y el refinamiento del ranking con propiedades no funcionales.

Para determinar el rendimiento del sistema en la recuperación de servicio con respecto al número de términos recuperados de la solicitud en lenguaje natural, se consideraron nueve propiedades no funcionales disponibles en el proceso de refinamiento del ranking y un banco de 200 servicios a partir de los 185 servicios reales, repitiéndose alguno de ellos por motivos de experimentales.

En la Figura 20, se muestra relaciones con tendencias lineales entre el número de términos usados para la recuperación de servicios y las fases de enriquecimiento de vocabulario y mapeo de datos al modelo del emparejador. Estos dos procesos son los que consumen más tiempo en la recuperación de servicios, ya que el enriquecimiento de vocabulario involucra operaciones complejas y dispendiosas, que involucran fuentes de conocimiento como WordNet y Flickr. Esto es evidente, dado que el trabajo con relaciones semánticas a partir de sistemas sociales como Flickr, requiere la conexión a Internet y por tanto, depende de la velocidad de conexión y el tiempo de respuesta del servidor de Flickr, para la obtención de los clústeres donde se encuentran los términos usados por los usuarios.

Por otro lado, el mapeo de datos involucra la ejecución de sentencias SQL, la adaptación de la *base de datos relacional* y los términos del usuario en entidades que permiten realizar las comparaciones por parte del emparejador, lo que requiere el manejo de arreglos de objetos; razón por la cual, esta funcionalidad también consume un tiempo considerable. A pesar de esto, reporta buenos resultados para el promedio de solicitudes realizadas por el usuario, ya que la recuperación de los servicios toma un tiempo inferior a 6,182 segundos cuando la solicitud presenta menos de 20 términos recuperados. Además, considerando que el sistema facilita la recuperación de diferentes tipos de recursos, que en muchos casos requieren intervenciones manuales, relacionadas por ejemplo, con la identificación de componentes para construir nuevos servicios dentro de un operador de telecomunicaciones, se concluye que los tiempos de rendimiento son adecuados.

La optimización en cuanto al rendimiento de estos procesos, no se abarca dentro del alcance del trabajo de grado, razón por la cual puede tenerse en cuenta para futuros proyectos de investigación. Por ejemplo, se puede considerar solo un enriquecimiento inicial antes de que el sistema sea ejecutado, trabajar con fuentes de conocimiento que no requieran conexiones a servidores, u optimizar la adaptación de datos con el emparejador.

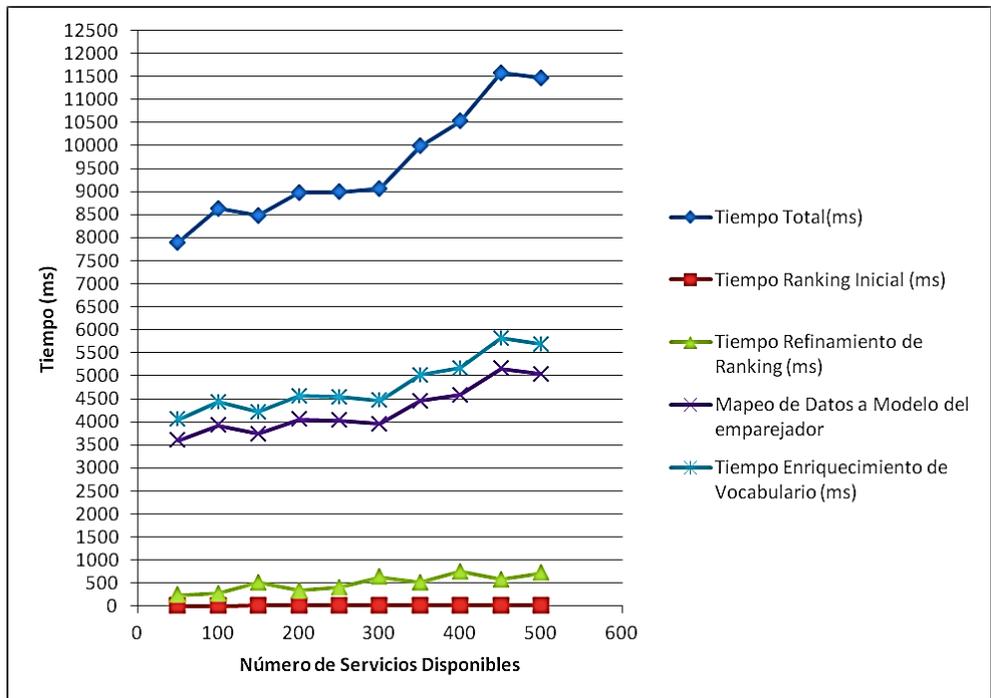


**Figura 20. Gráfica de rendimiento del sistema en la recuperación de servicios respecto al número de términos recuperados de la solicitud en lenguaje natural.**

En cuanto a los otros dos procesos (generación del ranking inicial y refinamiento del ranking), la generación del ranking inicial comparando los términos de entrada (solicitud del usuario) con las descripciones de los servicios presenta una disminución progresiva de rendimiento insignificante, lo que la convierte en una operación optimizada que aporta gran rendimiento al sistema, con un máximo de 45,5149 ms para la comparación con 100 términos. Un caso similar ocurre con el proceso de refinamiento del ranking con propiedades no funcionales, independiente del número de términos usados para la recuperación de servicios, presenta variaciones entre 297,999ms y 1139,742 ms, considerándola así una operación aceptable dentro del sistema.

Para determinar el rendimiento del sistema en la recuperación de servicio con respecto al número de servicios disponibles, se consideró una solicitud que permitía la recuperación de 40 términos y la selección del conjunto completo de las 9 propiedades no funcionales para el proceso de refinamiento del ranking.

En la Figura 21, se muestra un crecimiento paulatino del tiempo para las fases de enriquecimiento de vocabulario y el mapeo de datos al modelo del emparejador, al incrementar el número de servicios disponibles que pueden ser recuperados. Al igual que en el caso anterior, estos dos procesos son los que consumen más tiempo en la recuperación de servicios. Sin embargo, como se explicaba, reporta buenos resultados, agregando que en este caso, solo se presenta una disminución en el rendimiento, de menos de 2 segundos, con el incremento de 450 servicios en el banco de servicios, para cada proceso.



**Figura 21. Gráfica de rendimiento del sistema en la recuperación de servicios respecto al número de servicios disponibles.**

La generación del ranking inicial a través de comparaciones entre los términos de usuario y servicios, muestra ser una operación optimizada nuevamente, donde con 50 servicios disponibles el sistema realiza el proceso en tan solo 5,92 ms y con 500 servicios, lo realiza en menos de 14 ms. El proceso de refinamiento del ranking con propiedades no funcionales, presenta variaciones en este caso entre 246 y 750 ms, con tendencias de crecimiento leves, entre mayor sea la cantidad de servicios disponibles. Así, esta operación consume un tiempo aceptable, por lo que este proceso tampoco requiere mayores optimizaciones para trabajos futuros.

**Prueba de Calidad SC1:** Esta prueba permite medir la calidad de los rankings de servicios generados a partir de términos recuperados de la solicitud en lenguaje natural, además del desempeño del sistema para reorganizar dicho ranking con los valores de las propiedades no funcionales de los servicios. En las gráficas se presentan los resultados generales de las medidas de *precision*, *recall*, *overall*, *generalización de precisión fija promedio*, *ganancia acumulada normalizada media* y *ganancia acumulada descontada normalizada media*.

En la Figura 22, se presenta los resultados de las medidas de precisión del mecanismo de recuperación de servicios del proyecto con diferentes variaciones, es decir, realizando enriquecimiento de los términos de la solicitud solo con WordNet (*Basic-Ont*), con Flickr (*Basic-Folk*) y finalmente con las dos fuentes (*Basic-Full*). Adicionalmente, se muestran las medidas de precisión del emparejador base *WSColab*. De acuerdo a los resultados, el desempeño del sistema para recuperar servicios relevantes conforme al *benchmark* de referencia, es adecuado, y para la mayoría de entradas, se tienen niveles de precisión superiores al 90%. De las 5 entradas al sistema, *WSColab* tuvo mejores resultados de precisión que el prototipo realizado para 4 solicitudes, sin embargo 3 de estas medidas presentaron resultados muy similares entre *WSColab* y el prototipo propuesto. También se puede observar que la variación del prototipo con mejores resultados en *precision* es el enriquecido únicamente con relaciones semánticas provenientes de la fuente de conocimiento (Folcsonomías), lo cual se debe a que tiene una mayor cantidad de etiquetas comunes con las descritas en los servicios relevantes.

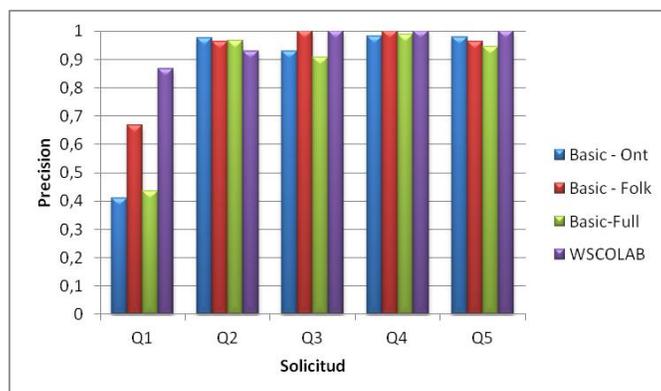


Figura 22. Medidas de Desempeño: *Precision* para sistemas de recuperación de servicios

En la Figura 23, se presenta los resultados de las medidas de *recall* del mecanismo de recuperación de servicios del proyecto con las variaciones descritas anteriormente y además las del emparejador base WSColab.

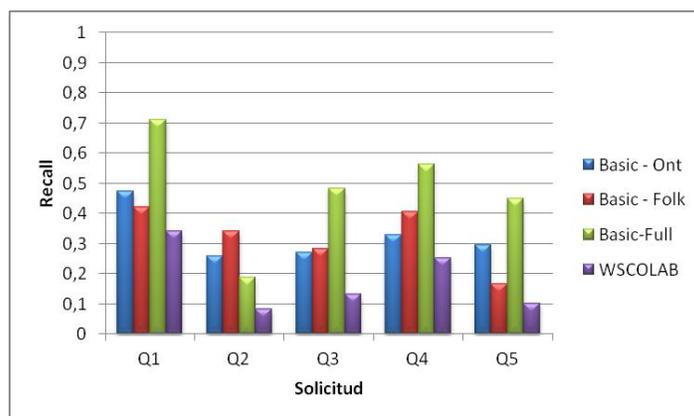


Figura 23. Medidas de Desempeño: *Recall* para sistemas de recuperación de servicios

Como muestran los resultados, a pesar de que WSColab tenía los mejores resultados de *precision*, seguidos de la variación del sistema con enriquecimiento de folcsonomías, los mejores resultados de *recall* son los obtenidos por el sistema con enriquecimiento de WordNet y Flickr, con un promedio de 50%. Esto indica una medida de *recall* no suficientemente aceptable, ya que se estarían descartando la mitad de los servicios relevantes. Esto se debe a las posibles variaciones de etiquetas usadas por usuarios para describir un servicio y las encontradas en la ontología léxica y en los clústeres de las folcsonomías. Sin embargo, la calidad de los resultados es mejor respecto al desempeño de sistemas similares de recuperación de servicios como es el caso del emparejador base (WSColab).

En la Figura 24, se presenta los resultados de las medidas de *overall* de los mecanismos de recuperación de servicios. Nuevamente el sistema de emparejamiento del proyecto muestra mejores resultados que el emparejador base (WSColab), lo que da al sistema mayor capacidad para agregar servicios relevantes y a su vez, descartar servicios no relevantes.

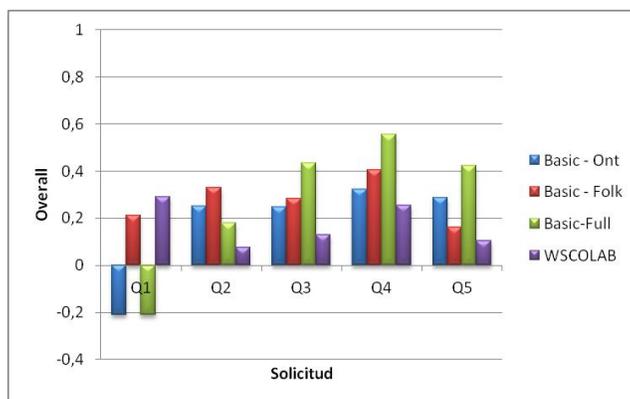


Figura 24. Medidas de Desempeño: Overall para sistemas de recuperación de servicios

En la primera entrada, se tienen valores negativos con la precisión por debajo del 50%, esto no significa que corresponda un valor insignificante y que el sistema tenga resultados de pésima calidad; lo que indica, para la primera entrada, es que el sistema de recuperación de servicios propuesto presenta una mayor correlación con la medida de *recall* en contraste a la de *precision*, como se indica en [149],

En la Figura 25, se ilustra la relación entre las medidas de *precision* y *recall* para el sistema general de recuperación de servicios. Se puede observar cómo el desempeño del sistema está en general ubicado en una zona de alta *precision* (superior al 90% a excepción de una entrada) y *recall* considerable (en promedio 48%), lo cual implica que el mecanismo es altamente selectivo y recupera un porcentaje considerable de servicios relevantes disponibles del banco de servicios.

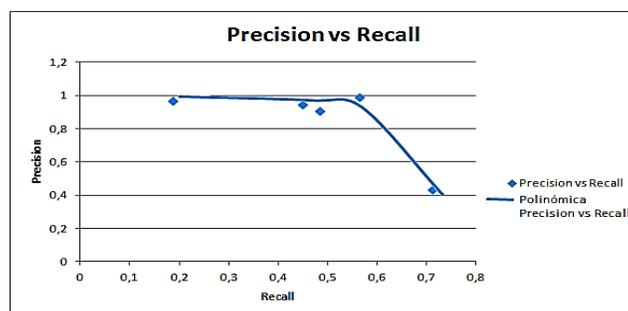


Figura 25. Gráfica de Precision vs. Recall para la recuperación de servicios

### Métricas No Convencionales de Evaluación

A continuación, se presentan los resultados aplicando las medidas definidas en [127], las cuales a diferencia de las métricas tradicionales de recuperación de información (*precision*, *recall* y *overall*) permiten, entre otras cosas, asignar un mayor grado de preferencia sobre ítems de mayor relevancia en un ranking. Estas métricas capturan el rendimiento del sistema en una única medida, considerando las métricas de *recall* y *precision* para su definición.

En la Figura 26, se presenta los resultados de las medidas de Ganancia Acumulada Normalizada Media (Averaged Normalized Cumulated Gain) de los mecanismos de recuperación de servicios. Puede confirmarse que el desempeño del sistema se encuentra ubicado en una zona alta (valores superiores al 35% hasta el 83%), superando al emparejador base (WSColab) y presentando pequeñas diferencias cuando es superado. Es de notar que el sistema con enriquecimiento únicamente de folcsonomías presenta mejores resultados (promedio del 64%), esto se debe a que los términos obtenidos de los clústeres de

Flickr tienen mayores coincidencias con las descripciones de servicios, que las relaciones de WordNet; por tanto, el desempeño del conjunto de WordNet con Flickr se ve comprometido.

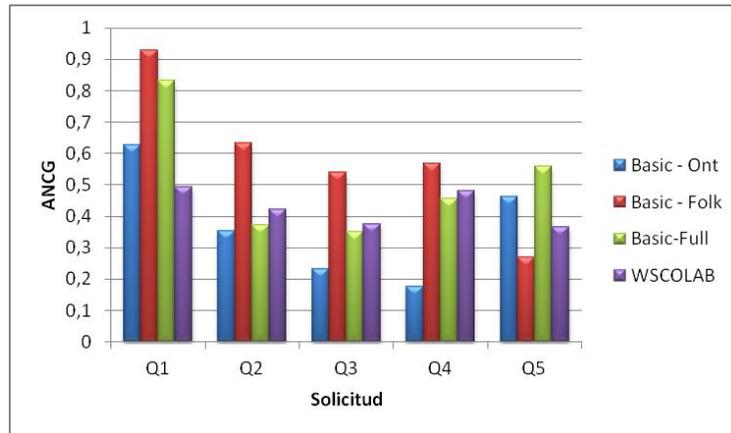


Figura 26. Medidas de Desempeño: Ganancia Acumulada Normalizada Media (ANCG) para sistemas de recuperación de servicios

La Figura 27, muestra los resultados de la versión simple de la medida anterior, es decir la Ganancia Acumulada Descontada Normalizada Media (Averaged Normalized Discounted Cumulated Gain). En este caso, sigue siendo notable que los mejores resultados corresponden al enriquecimiento con folcsonomías (promedio del 63%). El sistema de recuperación con las dos fuentes de conocimiento, tiene un promedio de 46% de desempeño, mientras que el emparejador base tiene un promedio del 51%. Esta medida indica que el sistema tiende a recuperar servicios altamente relevantes en las últimas posiciones del ranking, un poco más frecuente que el emparejador base, o dicho de otra manera, el emparejador base tiene un desempeño levemente mejor en recuperar servicios altamente relevantes en las primeras posiciones del ranking. A pesar de esto, como lo indican las figuras de *precision*, *recall* y *overall*, el emparejador base genera un mejor orden en el ranking de servicios recuperados, pero con un bajo desempeño a la hora de recuperar los servicios relevantes dados por el *benchmark* de referencia. En cambio, el sistema de recuperación del proyecto, recobra más servicios relevantes, aunque esto signifique tener un orden menos aproximado respecto al brindado por el emparejador base.

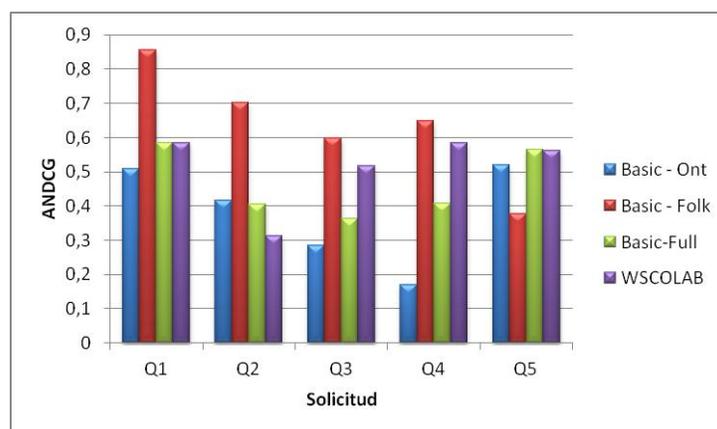
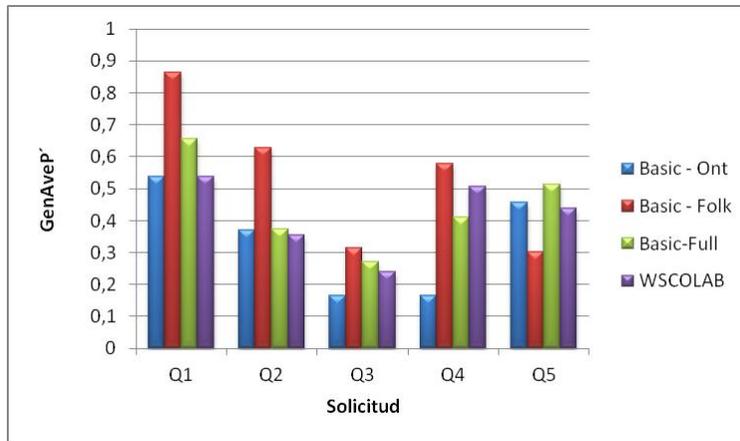


Figura 27. Medidas de Desempeño: Ganancia Acumulada Descontada Normalizada Media (ANDCG) para sistemas de recuperación de servicios

La Figura 28, muestra las medidas de Generalización de Precisión Fija Promedio (Fixed Generalization Average Precision). En el caso del enriquecimiento con folcsonomías se obtuvo un valor menor que los casos anteriores (promedio de 54%), mientras que el sistema de recuperación con las dos fuentes de conocimiento y el emparejador de base obtuvo promedios de 44% y 41% respectivamente.



**Figura 28. Medidas de Desempeño: Generalización de Precisión Fija Promedio (GenAveP) para sistemas de recuperación de servicios**

Los valores de esta medida son bajos y muestran un rendimiento regular del sistema de recuperación, esto se debe a que la ganancia acumulada obtenida del ranking generado por el sistema de recuperación es considerablemente menor a la ganancia acumulada del ranking ideal. La razón que también contribuye a valores de *recall* de 50%, es que el sistema para las diferentes entradas recupera una cantidad menor de servicios que el ranking ideal, razón por la cual la cantidad de ítems relevantes es reducida.

A pesar de mostrar resultados bajos en algunas de las medidas, se considera que el desempeño del sistema en general es adecuado. En primer, lugar mostró mejores resultados en la recuperación de servicios relevantes con las medidas de *precision*, *recall* y *overall* que el emparejador WSColab ganador del primer lugar en la categoría de evaluación cruzada del Tercer concurso internacional de selección de servicios semánticos, debido a su efectividad y tiempo corto de respuesta [150]. Por otra parte, los resultados bajos de las medidas de las gráficas 22,23,24,26,27,28 como se explicaba anteriormente, son debidas a que el *benchmark* de referencia fue producto de un promedio de los ranking definidos por 5 jueces expertos y no de un consenso entre ellos, por lo que el ranking ideal contiene muchos servicios relevantes que aportan en la ganancia acumulada ideal, disminuyendo por tanto, los resultados de estas medidas, las cuales dependen inversamente del valor de la ganancia acumulada ideal y directamente de la ganancia acumulada del sistema, generando rankings con una cantidad menor de ítems relevantes que el *benchmark* de referencia.

### Refinamiento del Ranking

En la Figura 29, se presentan los resultados correspondientes a las medidas de precisión del mecanismo de refinamiento de ranking con propiedades no funcionales, el cual tuvo ciertas variaciones, de acuerdo a la agrupación de dichas propiedades, tal como se muestra a continuación.

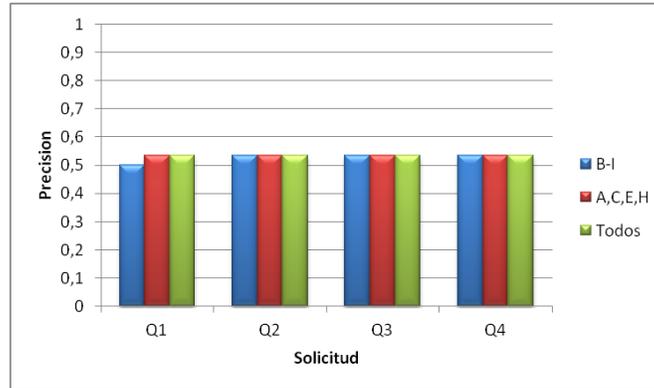
Propiedades No Funcionales usadas: **A.** *Tiempo de respuesta*, **B.** *Disponibilidad*, **C.** *Rendimiento*, **D.** *Habilidad de éxito*, **E.** *Confiabilidad*, **F.** *Cumplimiento*, **G.** *Mejores Prácticas*, **H.** *Latencia*, **I.** *Documentación*.

Grupos identificados:

- **Grupo I:** Propiedades B e I.

- **Grupo II:** Propiedades A, C, E y H.
- **Grupo III:** Propiedades A, B, C, D, E, F, G, H e I.

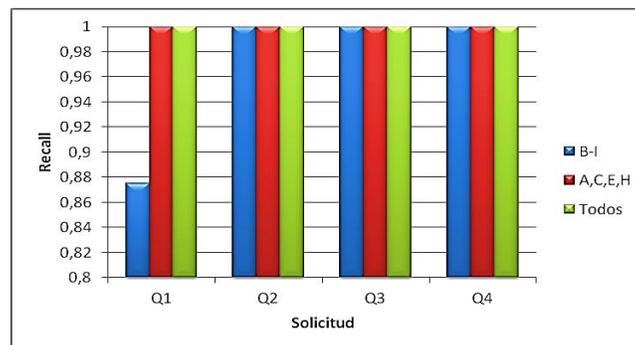
La selección de los grupos se realizó de forma aleatoria, con el fin de sustentar el hecho de que usuarios inexpertos seleccionarán propiedades no funcionales al azar.



**Figura 29. Medidas de Desempeño: Precision para el sistema de refinamiento de ranking**

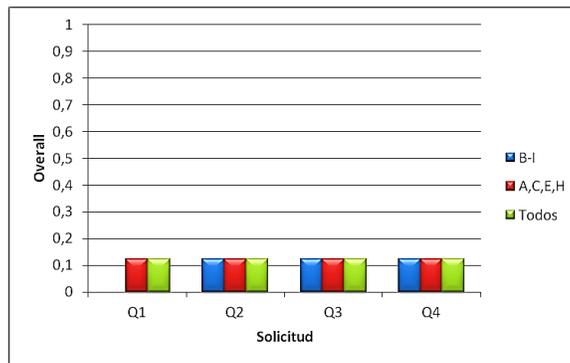
De acuerdo a los resultados obtenidos, el desempeño del sistema para recuperar servicios relevantes conforme con el *benchmark* de referencia no es el más óptimo, debido a que la mayoría de entradas tienen niveles de precisión inferiores al 55%. De esto se puede considerar que el sistema es poco selectivo; sin embargo, esto podría traducirse en beneficios, ya que el sistema no discrimina en alto porcentaje las propiedades no funcionales de los servicios a evaluarse y tiende siempre a brindar peso a todo el conjunto de PNF.

En la Figura 30, se presentan los resultados correspondientes a las medidas de *recall* del mecanismo de refinamiento de ranking. Como muestran los resultados, el *recall* en casi todas las solicitudes y para todos los grupos de PNF, mostró una magnitud de uno, lo que indica que el sistema recupera todos los servicios que se consideran importantes; sin embargo, y teniendo en cuenta la medida de precisión señalada anteriormente, se puede inferir que se obtienen mas servicios de los que se consideran relevantes dentro de la selección. Esto es debido a que el excedente de servicios identificados puede contener PNF óptimas, las cuales no están siendo tenidas en cuenta directamente por el *benchmark* de referencia.



**Figura 30. Medidas de Desempeño: Recall para el sistema de refinamiento de ranking**

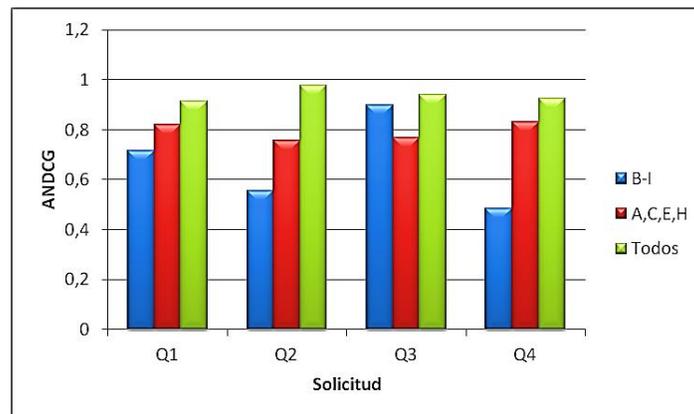
En la figura 31, se presentan los resultados de las medidas de *overall* de mecanismos de refinamiento de ranking. Nuevamente el sistema de refinamiento muestra unos resultados no muy altos, lo que significa que la selección de los servicios de acuerdo al *benchmark* de referencia no es tan exacta, permitiendo que algunos servicios con propiedades no funcionales relevantes no se destaquen sobre los demás.



**Figura 31. Medidas de Desempeño: Overall para el sistema de refinamiento de ranking**

A continuación, y de acuerdo a las métricas definidas en [127], se presenta otro tipo de medida expuesta a continuación.

La Figura 32, muestra los resultados de la Ganancia Acumulada Descontada Normalizada Media (Averaged Normalized Discounted Cumulated Gain), la cual obtuvo buenos resultados en la mayoría de las solicitudes realizadas al sistema. Cabe resaltar, que la mejor selección de servicios para todos los casos se da cuando el número de propiedades no funcionales a evaluar está compuesto por la totalidad de las propiedades, hecho que señala la posibilidad que tiene el sistema para brindar peso igualitario a todas las PNF y no necesariamente a un conjunto reducido de las propiedades.



**Figura 32. Medidas de Desempeño: Ganancia Acumulada Descontada Normalizada Media (ANDCG) para el sistema de refinamiento de ranking**

En conclusión, la evaluación del sistema de refinamiento exhibe una tendencia del sistema a no brindar una relevancia destacada a los servicios que contienen pocas PNF óptimas; sino, por el contrario, señala la posibilidad de identificar como relevantes a aquellos servicios que poseen óptimas condiciones en todas o la mayoría de las PNF, hecho que demuestra la importancia que tiene el refinamiento y la autonomía que presenta el sistema para considerar importantes a todas las PNF existentes de cada servicio.

### 5.3. RESUMEN

En este capítulo se abordó una descripción detallada del proceso de desarrollo llevado a cabo, alrededor de la construcción de un sistema software que implementa los mecanismos propuestos para la recuperación de servicios en entornos convergentes a partir de solicitudes en lenguaje natural, considerando propiedades funcionales y no funcionales, tanto de los servicios como de la petición del usuario. Además, del

establecimiento de los requerimientos funcionales del prototipo, fue posible definir los casos de uso del sistema y la arquitectura de referencia que permitió soportar el desarrollo del mismo.

Por otro lado, se describieron las metodologías de evaluación utilizadas para establecer el desempeño de los mecanismos del sistema. Las metodologías definidas tomaron como base, además del proceso de *Benchmarking*, diversas métricas (de Recuperación de Información y no convencionales) para determinar la calidad de los mecanismos propuestos.

De acuerdo con estas medidas, el desempeño general del sistema fue satisfactorio cuando se recuperan servicios a partir de solicitudes realizadas en lenguaje natural. Igualmente el mecanismo de refinamiento propuesto tuvo resultados positivos reorganizando el ranking con relación a preferencias no funcionales seleccionados por el usuario.

De esta manera, se demostró la importancia de la creación de sistemas orientados al usuario final, que permitan recuperar diferentes tipos de recursos para la composición personalizada de servicios, para este caso servicios Web y de Telecomunicaciones. De igual manera, se mostró la importancia de las diferentes tareas de PLN para el correcto tratamiento de solicitudes flexibles realizadas en inglés y adicionalmente, cómo a través de un mecanismo sintáctico con procesos de enriquecimiento semántico y etiquetado colaborativo, es posible mejorar el emparejamiento de servicios, al mismo tiempo que optimiza los tiempos de ejecución del sistema.

Finalmente, se detalló un mecanismo efectivo para considerar criterios no funcionales dentro del proceso de generación de un ranking final de servicios y cómo, a partir de términos contenidos en las solicitudes del usuario, se puede establecer el orden de ejecución de los servicios recuperados. Todo esto evidencia la viabilidad, favorabilidad y por ende, la aprobación de las hipótesis planteadas en el presente trabajo de grado.

## 6. CONCLUSIONES Y TRABAJO FUTURO

El promisorio panorama en la generación de recursos provistos sobre entornos convergentes (Web + Telecomunicaciones), enfocada en el usuario final, evidencia en la actualidad una tendencia interesante, donde usuarios con limitadas habilidades técnicas, no solo consumen recursos, si no que tienen también la capacidad de crearlos. Por lo general, esta característica es más común en la Web, gracias al principio fundamental de SOA (Service Oriented Architecture): el reuso de componentes existentes.

Esto conlleva a que los usuarios requieran de diferentes técnicas y/o herramientas con el fin de diseñar y crear servicios personalizados dentro del entorno convergente (actividad habitualmente llevada a cabo por desarrolladores expertos). Es por esta razón que se hace necesaria la existencia de medios que faciliten el desarrollo de las tareas de diseño y creación de servicios compuestos, sin necesidad de poseer conocimientos elevados en la materia.

Para este propósito, fue abordado en el presente trabajo de grado, un mecanismo que permite a los usuarios expresar sus solicitudes de manera más simple, obteniendo como resultado un flujo aproximado de servicios, directamente relacionados con los requerimientos iniciales de una solicitud. Esta recuperación, fue realizada a través de un mecanismo que permite a los usuarios considerar requerimientos funcionales y no funcionales de diversos servicios, usando un lenguaje entendible facilitando así su uso.

El mecanismo mencionado, está basado en algunos trabajos de investigación relacionados con el procesamiento de lenguaje natural y algunas técnicas de emparejamiento usadas para la generación de rankings específicos, lo cual permitió establecer procesos que facilitaron la generación de un ranking de servicio.

Este capítulo describe inicialmente las principales contribuciones del trabajo realizado, posteriormente presenta las conclusiones a las que se llegó, y finalmente se proponen algunos trabajos futuros.

### 6.1. CONTRIBUCIONES

Dentro de las principales contribuciones del presente proyecto de grado se destacan las siguientes:

- Establecimiento de una tubería (*pipeline*) de análisis de lenguaje natural, donde se definieron 6 tareas que se ejecutan secuencialmente (tokenización, filtrado de palabras, etiquetado de palabras, clasificación de nombres de entidad, desambiguación semántica y lematización). Este conjunto de tareas y la adaptación dentro de un *pipeline*, fue desarrollado a partir del uso de una plataforma orientada a la ingeniería de texto denominada GATE, la cual tiene dentro de sus características algunas de las tareas nombradas a las cuales se les sumaron tareas de otros trabajos de investigación, como el desambiguador semántico.
- Dentro del establecimiento del proceso de selección de servicios a partir de propiedades no funcionales, se realizó una adaptación de un mecanismo de enriquecimiento de palabras, tanto de los términos usados por los usuarios al momento de ingresar una solicitud, como a las etiquetas que describen funcionalmente los servicios. El mecanismo permite expandir dichos términos, con el fin de solucionar los problemas de extensión y amplitud que tiene el lenguaje, ya que para describir un mismo elemento pueden utilizarse diferentes palabras, lo que conlleva a la inclusión de una amplia cantidad de palabras. Para ello fueron consideradas dos fuentes de conocimiento, por un lado una fuente de conocimiento formal como es la ontología de Wordnet (contiene hiperónimos, hipónimos, holónimos, merónimos) y por otro lado se encuentra la fuente de origen social Flickr (palabras dentro de un cluster contienen relaciones semánticas de tipo social) que permite complementar la primera en la expansión de términos.
- Definición de técnicas y algoritmos de recuperación automática de servicios. El enfoque propuesto, permite adaptar la recuperación de servicios a cualquier tipo de recurso consumible (widgets, APIs,

aplicaciones, etc), lo que lo hace adecuado a las características y requisitos propios de entornos convergentes. Revisando diferentes soluciones sobre recuperación y búsqueda de servicios, se consideraron los servicios como recursos y de esta manera, definir los mecanismos necesarios alrededor de estos para su recuperación y selección.

- Desarrollo de un sistema automático para dispositivos móviles, donde se puede configurar los diferentes factores definidos en la monografía e ingresar solicitudes de un servicio expresadas en inglés de un servicio esperando recomendaciones y diagramas de los bloques de servicios recuperados de acuerdo a la solicitud. El prototipo de escritorio, puede considerarse semiautomático, ya que los desarrolladores inexpertos tienen mayor conocimiento de propiedades no funcionales que los usuarios finales, con lo cual pueden seleccionar criterios de preferencia sobre las propiedades no funcionales, sin embargo, si el desarrollador no lo desea, el sistema genera un resultado sin su intervención.
- La definición de un mecanismo de reordenamiento de un ranking a partir del análisis de las propiedades no funcionales. Esto fue desarrollado a partir de la definición de un factor de relevancia ingresado por el usuario, que tiene en cuenta el peso asignado a las propiedades no funcionales señaladas por los usuarios, respecto al total de propiedades no funcionales existentes en el repositorio. Adicionalmente el mecanismo contiene una técnica que permite establecer el factor de relación entre las diferentes propiedades evaluadas (medición que expresa el nivel de preferencia en que dos o más propiedades no funcionales deberían ser consideradas con un mismo nivel de relevancia), con el fin de obtener valores de mejor calidad asignados a cada una de las propiedades.
- La generación de una aproximación de un flujo de control genérico, se convierte en una de las contribuciones más importantes del presente trabajo, debido a la poca documentación y a la implementación de técnicas complejas en la elaboración de este tipo de procesos. El mecanismo propuesto se basa en la abstracción de palabras de una solicitud realizada en lenguaje natural, las cuales definen elementos de orden o secuencia como palabras de conjunción o números ordinales. Con estos elementos se realizó una aproximación del orden de ejecución de los elementos dentro de la solicitud, mostrando al usuario la representación de estructuras o diagramas abstractos de grafos con el uso de una herramienta de software gráfica.
- La recopilación de cerca de ciento ochenta servicios Web y de Telecomunicaciones obtenidos de diversos repositorios, los cuales fueron etiquetados con propiedades funcionales y no funcionales. Las propiedades funcionales relacionan las interfaces de comportamiento y entrada/salida de un servicio, mientras que las propiedades no funcionales relacionan propiedades como: documentación, disponibilidad, tiempo de respuesta, rendimiento, latencia, mejores prácticas, conformidad y confiabilidad.
- El prototipo SeRTool (Service Request Based on Natural Language Tool) se constituye en una herramienta automática para que diversos usuarios puedan formular solicitudes de servicios en lenguaje natural, tanto desde su dispositivo móvil como desde su ordenador, solicitudes en lenguaje natural con diversas necesidades de servicios. La herramienta a través de los mecanismos expuestos anteriormente, obtiene como salida una recomendación con un conjunto de servicios, considerados como relevantes para el usuario, los cuales se encuentran ordenados de forma aproximada. Este prototipo reduce la complejidad y disminuye el tiempo que habitualmente usuarios expertos emplean en el proceso de recuperación de servicios.

## **6.2. CONCLUSIONES**

Dentro del proyecto se abordaron las temáticas de análisis de lenguaje natural y emparejamiento sintáctico con enriquecimiento semántico, entre etiquetas descriptoras de servicios y palabras usadas por los usuarios finales, en la formulación de consultas de servicio. El primer tema se desarrollo a partir del análisis de tareas para el procesamiento de lenguaje natural, en el cual se excluye el uso de plantillas en las solicitudes

de los usuarios, como se realiza en muchos de los trabajos de procesamiento de lenguaje natural orientados a procesos de recuperación de servicios. Adicionalmente se analizaron varias técnicas de emparejamiento, que empleaban tanto de etiquetas simples, como ontologías y folksonmias. Este análisis, permitió proponer un mecanismo de comparación sintáctica basado en el uso de etiquetas enriquecidas semánticamente, entre los conceptos que definen los servicios y las palabras usadas por los usuarios finales.

Al finalizar este proyecto se entregó un prototipo denominado SeRTool (Service Request Based on Natural Language Tool), el cual se basa en el mecanismo antes mencionado, el cual permite la obtención de un ranking de servicios y un flujo de su posible ejecución. Este prototipo permite recomendar a los usuarios diversas salidas del sistema que han sido almacenadas previamente, lo cual disminuye notoriamente el tiempo de procesamiento y los recursos usados.

Lo anterior, permite concluir que el trabajo realizado, facilita la recuperación de servicios, a partir del ingreso de solicitudes realizadas por usuarios finales en lenguaje informal, teniendo en cuenta propiedades funcionales y no funcionales de los servicios.

A continuación se describen las principales conclusiones que surgieron durante la ejecución del presente proyecto:

- A partir del estudio de diversas técnicas, se pudo deducir que la implementación de técnicas de análisis de lenguaje natural para procesos de selección de servicios, facilita los procesos de recuperación de servicios, reduciendo el tiempo requerido, la complejidad y en este sentido ampliando el número de usuarios aptos para este tipo de procesos.
- La implementación de un sistema de recomendación y autocompletado, facilita una interacción con el usuario, para obtener el flujo final del sistema, ya que se inducen elementos formales dentro de una solicitud informal, sin necesidad de establecer restricciones o limitaciones en la captura de requerimientos.
- El enriquecimiento semántico de palabras de los usuarios y las etiquetas descriptoras de un servicio, permiten que el proceso de emparejamiento sintáctico sea más preciso, ya que cuando los usuarios ingresan palabras en una solicitud, muchas veces éstas no coinciden con los términos que describen los servicios en el repositorio. Por este motivo, al realizar un enriquecimiento semántico de la solicitud y de las descripciones de servicios, a través de diversas fuentes de conocimiento, permite que el emparejamiento sintáctico tenga mayor alcance, sin la necesidad de que las palabras introducidas por los usuarios coincidan exactamente con las usadas para la descripción de los servicios.
- La inclusión de propiedades no funcionales en el proceso de recuperación de servicios, permite que los servicios seleccionados, no solo sean los más relacionados a las propiedades funcionales expuestas en lenguaje natural por los usuarios, sino también garantiza que estos servicios sean contengan propiedades con buenas características en cuanto a criterios de disponibilidad, tiempo de respuesta documentación entre otras. Lo anterior deriva en un mejor desempeño en ejecución por parte de los servicios recuperados y de ese modo a una mayor satisfacción en los usuarios finales.
- Las pruebas de rendimiento realizadas sobre todo el sistema muestran un crecimiento exponencial, respecto al número de términos en la solicitud de entrada y el tiempo de respuesta del sistema (recuperación de servicios y flujo de control). Tomando como punto de referencia, que la mayoría de solicitudes expresadas en lenguaje natural se encuentran alrededor de 20 palabras por solicitud; el sistema en este caso tarda alrededor de un segundo (dependiente de la red de acceso a internet para el enriquecimiento con Flickr) para dar respuesta a la solicitud, tiempo suficiente y altamente satisfactorio teniendo en cuenta la magnitud de los procesos llevados a cabo al interior del sistema.
- Basándose en los mecanismos de emparejamiento propuestos para el desarrollo del emparejador WSColab y el uso de tareas de PLN optimizadas por GATE, fue posible conseguir tiempos ágiles de

respuesta del sistema. Sin embargo es evidente que existen aspectos por mejorar con el fin de reducir los tiempos de respuesta del sistema al mínimo posible.

- Las pruebas de calidad de la salida del sistema mostraron que el porcentaje más alto de conformidad fue en la selección de términos del flujo de control con un valor de 91,6% y el menor nivel de conformidad, se presentó entre los términos de comportamiento con un valor de 68,1%. Por otro lado el número de servicios identificados acordes a las solicitudes, mostró un fuerte nivel de satisfacción con un 86,1%, mientras que los términos identificados por servicio mostraron una satisfacción del 62,9%. Por lo anterior se puede deducir que frente a una solicitud informal, se puede obtener una estructura de salida de buena calidad..
- En cuanto a la evaluación del sistema de refinamiento, se puede afirmar que los resultados aunque no muy buenos a nivel numérico, si representan una condición del sistema que brinda a todos los servicios un peso relevante, con el objetivo de que el refinamiento tenga en cuenta no solo aquellos elementos seleccionados por el usuario, sino, en general tenga en cuenta todos los valores de las PNF y de esta manera pueda llegar a tener servicios con mejores PNF.
- En cuanto a cada una de las tareas de procesamiento de lenguaje natural, se puede mencionar que la desambiguación de sentidos de palabras fue la tarea que presentó menor calidad con un porcentaje menor al 50% en precisión y recall, mientras que el etiquetado de palabras fue el que presentó un mejor nivel de precisión y recall con un porcentaje de 80%. Es de notar que el desempeño del sistema para realizar tareas de desambiguación, no resulta tan adecuado al usar palabras comunes en varios contextos, sin embargo, en la solicitud de los usuarios generalmente los términos tienen una relación estrecha con un determinado contexto el cual da mayor relevancia a un sentido específico de los que puede tener una palabra.
- En cuanto a la calidad de los resultados en la generación inicial de ranking de servicios, se puede observar que el desempeño del sistema es de alta precisión con un porcentaje superior al 90% y un recall moderado con un promedio 48%, lo que implica que el mecanismo de recuperación es altamente selectivo, obteniendo porcentajes considerables de servicios relevantes, a partir del conjunto de servicios disponibles. Por otro lado, la baja medida de recall no es tan aceptable ya que se estarían descartando la mitad de los servicios relevantes, lo cual se debe a las variaciones en las palabras usadas por los usuarios para describir un servicio y las encontradas en la ontología léxica y en los clústeres de las folcsonomías. Sin embargo, la calidad de los resultados es mejor respecto al desempeño de sistemas similares de recuperación de servicios como el emparejador WSColab.

### **6.3. TRABAJOS FUTUROS**

El presente trabajo puede considerarse como un estudio inicial de un mecanismo de recuperación automática de servicios en entornos convergentes a partir de solicitudes realizadas en lenguaje natural por diversos tipos de usuarios. Este trabajo aporta soluciones al problema de descubrimiento de servicios, emparejamiento sintáctico de términos y generación aproximada de un flujo de control. De esta manera, en el campo de investigación del presente proyecto de grado, se proponen los siguientes trabajos futuros:

#### **Inclusión de métodos de auto aprendizaje en el proceso de análisis de lenguaje natural.**

La fase de análisis de lenguaje natural propuesta, contiene los elementos necesarios y suficientes para llevar a cabo una correcta identificación de términos relevantes a partir de una solicitud. Sin embargo la literatura muestra que diversos trabajos [4], [55] han señalado que la inclusión de métodos de aprendizaje, permiten aumentar la exactitud en estos procesos de identificación, lo cual puede abrodarse en un proyecto de investigación futuro.

## **Perfeccionamiento del proceso de generación y elaboración del flujo de control**

El mecanismo propuesto en el presente proyecto, tiene como propósito identificar un flujo de control genérico a partir del análisis de elementos dentro de la solicitud realizada en lenguaje natural. Esta generación tiene en cuenta las relaciones de orden y conjunción que existen dentro de cada petición y a partir de ello se genera el flujo respectivo. Sin embargo, para un análisis posterior, podría tenerse en cuenta elementos como las interfaces de los servicios (Elementos de Entrada y Salida) para la generación del flujo, obteniendo así una mayor relación entre sus componentes y de la misma forma obteniendo una mejor elaboración del servicio compuesto.

## **Extensión del repositorio de servicios, etiquetados con propiedades no funcionales.**

La recopilación de ciento ochenta servicios Web y de Telecomunicaciones realizada en el presente trabajo, contiene un conjunto de propiedades no funcionales asociadas a cada uno de ellos, las cuales fueron tomadas de diversos repositorios existentes en la Web. En este sentido, es conveniente aumentar tanto el número de servicios dispuesto en el repositorio, como el número de propiedades no funcionales asociadas a cada servicio. Esto con el fin de garantizar y asegurar la entrega de buenos servicios con cada vez mejores características para los usuarios.

## **Optimización de mecanismos de recuperación de servicios.**

A pesar de que el sistema mostró un buen desempeño en rendimiento, es posible realizar algunas modificaciones en cuanto a los tiempos de ejecución de algunos módulos, que toman un tiempo considerable para obtener su respuesta. Se sugiere principalmente que dentro de la fase e lenguaje natural, sean optimizados los módulos de desambiguación semántica y el modulo de reconocimiento de entidades, ya que sus tiempos de ejecución repercuten directamente con el tiempo total de respuesta frente a una solicitud de entrada. Para ello, se debe considerar algunas técnicas que permitan ejecutar reglas, para el reconocimiento de entidades que se ajusten a un conjunto de parámetros mayor al señalado en el presente trabajo, asimismo, lo que concierne a la desambiguación semántica, puede hacerse uso de alguna otra extensión del Algoritmo de Lesk usado, con el fin de mejorar la exactitud y tiempo de desambiguación mostrado.

## **Extensión a otro tipo de recursos**

Debido a la facilidad del enfoque propuesto para adaptar la recuperación de servicios a cualquier tipo de recurso consumible (widgets, APIs, aplicaciones, etc), es posible mejorar la implementación del sistema con la extensión a otro tipo de recursos. De esta manera se puede aumentar el nivel de satisfacción del usuario.

## **Experimentación del prototipo en un ambiente real.**

Por último, se propone realizar la experimentación del prototipo en un ambiente real con múltiples usuarios reales, para determinar la respuesta tanto por parte de los usuarios como del sistema, y validar así su desempeño en actividades de recuperación con un proveedor de servicios real.

## REFERENCIAS

- [1]. V. K. Gurbani and X.-H. Sun, "*Internet Telephony: The Evolution to a Service-Oriented Architecture*," in *Architecting the Telecommunication Evolution: Toward Converged Network Services*, Darien: Auerbach Publications, 2006, p. 284.
- [2]. ITU-T, "General overview of NGN," in *Scope and Propose* vol. Y.2001: ITU-T, 2004, p.10.
- [3]. G. Bond, et al. "*Unified Telecom and Web Services Composition: Problem Definition and Future Directions*", In Proc. of (IPTCOMM '09). Atlanta, Georgia, USA, 2009.
- [4]. F.-C. Pop, et al. "*Natural Language based On-demand Service Composition*," *International Journal of Computers, Communications & Control*, vol. V, pp. 871-883, 2010.
- [5]. A. Bandara, "Semantic Description and Matching of Services for Pervasive Environments," Doctor of philosophy Semantic Description and Matching of Services for Pervasive Environments, Engineering, Science and Mathematics, Universidad de Southampton. Southampton, 2008.
- [6]. J. C. Corrales, "Behavioral matchmaking for service retrieval," Doctor of Philosophy in Sciences, Computer Science, University of Versailles Saint-Quentin-en-Yvelines, Versailles, 2008
- [7]. X. Yang, "A Framework for Semantic Service Discovery." Cambridge: MIT Oxygen Alliance, 2001, pp. 1-2.
- [8]. M. Multin, "Semantic Service Discovery using Natural Language Queries," Diploma, Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, 2009.
- [9]. K. Sutthikulphanich, "A Demonstration on Service Compositions based on Natural Language Request and User Contexts," Master of Science in Communication Technology, Telematics, Norwegian University of Science and Technology, Trondheim, 2008.
- [10]. E. Al-Masri and Q. Mahmoud, "*Discovering the Best Web Service: A Neural Network-based Solution*," presented at the IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA, 2009.
- [11]. R. N. a. B. Lee, "*Web Service Discovery with additional Semantics and Clustering*," *International Conference on Web Intelligence*, vol. 1, p. 5, 2007.
- [12]. E. Al-Masri and Q. Mahmoud (2008), The QWS Dataset [Online]. Disponible en <http://www.uoguelph.ca/~qmahmoud/qws/index.html> [Accedido Ene. 18, 2012].
- [13]. J. Community. (2007). "*JSR 289: SIP Servlet v1.1*," [Online]. Disponible en: <http://jcp.org/en/jsr/detail?id=289>. [Accedido Sept. 18, 2011].
- [14]. T. Dinsing, et al. "*Service composition in IMS using Java EE SIP servlet containers*," *Ericsson Review* pp. 92-96, 2007.
- [15]. E. C. Pedraza, et al. "Automatic service retrieval in converged environments based on natural language request," in *SERVICE COMPUTATION 2011*, 978-1-61208-152-6, (Rome, Italy), pp. 52–56, September 25 201.
- [16]. Prud'hommeaux. E, "SPARQL Query Language for RDF," W3C Recommendation. 2008. [Online]. Disponible en: <http://www.w3.org/TR/rdf-sparql-query/> [Accedido Nov. 18, 2011].
- [17]. D. Lozano, et al. "*WIMS 2.0: Converging IMS and Web 2.0. Designing REST APIs for the Exposure of Session-Based IMS Capabilities*," presented at the Proceedings of The Second International Conference on Next Generation Mobile Applications, Services, and Technologies, 2008.
- [18]. D. Moro, et al. "WIMS 2.0: Enabling Telecom Networks Assets in the Future Internet Of Services," in Proc. of (*ServiceWave '08*), pp. 74-85. 2008.
- [19]. ITU-T, "Converged services framework functional requirements and architecture," in *Definitions* vol. 2013: ITU-T, 2006, p. 39.
- [20]. ITU-T, "ITU-T Y.2000 series – Supplement on service scenarios for convergence services in a multiple network and application service provider environment," in *Service requirements and capabilities for convergence services* vol. 2013: International Telecommunication Union - T, 2008, p. 20.
- [21]. J. A. E. Sanches. "*Open Platform for User-centric service Creation and Execution*", 2006. [Online]. Disponible en: [ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate\\_d/st-ds/opuce-project-story\\_en.pdf](ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/st-ds/opuce-project-story_en.pdf). [Accedido: Dic. 10, 2011].
- [22]. C. Cordier. "*Service Platform for Innovative Communication Environment*," 2009. [Online]. Disponible en: [http://www.ist-spice.org/documents/project\\_summary\\_SPICE.pdf](http://www.ist-spice.org/documents/project_summary_SPICE.pdf). [Accedido: Ene. 30, 2012].

- [23]. Ericsson. "Ericsson Composition Engine – Next-generation IN," 2009. [Online]. Disponible en: [http://www.ericsson.com/ericsson/corpinfo/publications/review/2009\\_02/files/NGIN.pdf](http://www.ericsson.com/ericsson/corpinfo/publications/review/2009_02/files/NGIN.pdf) [Accedido: Feb. 10, 2012].
- [24]. ITU-T, "Enhanced Telecom Operations Map (eTOM) –The business process framework," in *The eTOM business process framework* vol. M.3050.1: ITU-T, 2004, pp. 8-35.
- [25]. H. Jiejin, "A Practical Approach to the Operation of Telecommunication Services driven by the TMF eTOM Framework," Magister, Teoria del Senyal i Comunicacions, Universitat Politécnica de Catalunya, Catalunya, 2009.
- [26]. TMF. "Information Framework (SID) In Depth," 2005. [Online]. Disponible en: <http://www.tmforum.org/InformationFramework/6647/home.html>. [Accedido: Feb. 15, 2012].
- [27]. A. Maaradji, et al. "N. Social-Based Web Services Discovery and Composition for Step-by-Step Mashup Completion," In Proc. of (ICWS'11), pp. 700-701. 2011.
- [28]. P. Leedy and J. Ormrod. *Practical Research: Planning and Design*, 8th Ed. vol 1. Prentice Hall. 2004.
- [29]. E. D. Liddy, "Natural Language Processing for Information Retrieval & Knowledge Discovery," *Text Wise, 1st Ed.* vol. 1, 2001, pp 10.
- [30]. B. Manaris. "Natural Language Processing: A Human–Computer Interaction Perspective" *Advances in Computers*. M. V. Zelkowitz, ed.1, vol. 47, pp. Academic Press, New York, 1998, pp.1-66.
- [31]. "What is a rule-based system?". Feb. 16, 1996. [Online]. Disponible en: <http://www.jpaine.org/students/lectures/lect3/node5.html>. [Accedido: Mar. 20, 2012].
- [32]. G. Kumar. "Artificial neural network and its applications," *I.A.R.I.* 2005.
- [33]. Gawinecki, M., et al. "WSCOLAB: Structured for Collaborative Tagging Web Service Machmaking". In Proc. of. (S3'09), pp 1-8. 2009.
- [34]. M. Magableh, "A Generic Architecture for Enhanced Semantic Tagging Systems," Doctor Thesis. Leicester - United Kingdom July 2011.
- [35]. T. Gruber, "A translation approach to portable ontology specifications. *Knowledge Acquisition*". *Knowl. Acquis*, vol 5, no 2, pp. 199-220, Jun. 1993.
- [36]. L. Ordoñez. "comparación semántica de tareas entre dos procesos de negocio de telecomunicaciones". *Tesis de Grado, Universidad del Cauca, Popayán-Cauca.* 2011.
- [37]. F. Abbattista, et al, "Shaping personal information spaces from collaborative tagging systems", in Proc. of. (KES'07/WIRN'07), vol. 4694, pp. 728-735, Heidelberg: Springer (Germany). 2007.
- [38]. E. Quintarelli, "Folksonomies: power to the people", June. 2005 [Online]. Disponible en: <http://www.iskoi.org/doc/folksonomies.htm>. [Accedido: Ago. 16, 2011].
- [39]. J. Cassel. "What Are Taxonomies?,". *Accidental Taxonomies*. [E-book]. Disponible en: <http://www.hedden-information.com/Accidental-Taxonomist-Chapter-1.pdf>
- [40]. C. Kiu and E. Tsui. "TaxoFolk: A hybrid taxonomy - folksonomy classification for enhanced knowledge navigation". *Pre-Published Version.* 2007 [Online]. Disponible en: <http://repository.lib.polyu.edu.hk/jspui/bitstream/10397/4595/1/KMRP%20paper.pdf> [Accedido: Mar. 26, 2012].
- [41]. D. Fichter "Intranet applications for tagging and folksonomies," Jun 2006 [Online]: Disponible en: [http://faculty.philau.edu/kayk/kkay/articles/Tagging\\_folksonomy.pdf](http://faculty.philau.edu/kayk/kkay/articles/Tagging_folksonomy.pdf) [Accedido: Abr. 17, 2012].
- [42]. OECD Minister. "Convergence and Next Generation Networks". 2007 [Online]. Disponible en: <http://www.oecd.org/internet/interneteconomy/40761101.pdf> [Accedido: Mar. 12, 2012].
- [43]. E.C. Pedraza. "User-driven Service Retrieval Platform for Converged Environments". *Natural Language Processing and Information Systems*, vol 7337, pp 203-209. 2012.
- [44]. A. Fernández, et al. "Closing the Service Discovery Gap by Collaborative Tagging and Clustering Techniques". In SMRR, *CEUR Workshop Proceedings*, vol 416, 2008
- [45]. K. Sutthikulphanich, "A Demonstration on Service Compositions based on Natural Language Request and User Contexts," Master of Science in Communication Technology, Telematics, Norwegian University of Science and Technology, Trondheim, 2008.
- [46]. M. S. D. Richards, S. van Splunter and F.M.T. Brazier, "Artificial Intelligence: a Promised Land for Web Services " *Intelligent Interactive Distributed Systems*, vol. 1, p. 6, 8 March 2004.
- [47]. Sims. (2006). *Semantic Interfaces for Mobile Services*. [Online]. Disponible en: <http://www.ist-sims.org>.
- [48]. IBM. "What is Watson?" 2011. [Online]. Disponible en: <http://www-03.ibm.com/innovation/us/watson/what-is-watson/index.html> [Accedido: Oct. 11, 2011].

- [49]. M. C. McCord. "Using Slots and Modifiers in Logic Grammars for Natural Language". *Artificial Intelligence*. Vol 18, May 1982, pp. 327–367.
- [50]. J. Lu. "Prolog a Tutorial Introduction" Computer Science Department Bucknell University Lewisburg. 2002.
- [51]. A. Schmidt & N. Skamriis, "Textual Similarity: Comparing texts in order to discover how closely they discuss the same topics", In Proc. of (IMMBSc-2008), Kongens Lyngby 2008.
- [52]. C. Seon, et al, "Named Entity Recognition using Machine Learning Methods and Pattern-Selection Rules", In Proc of (Seo 2001), pp 1-8. 2001.
- [53]. D. Nadeau and S. Sekine. "A survey of named entity recognition and classification" 2006 [Online]. Disponible en: <http://nlp.cs.nyu.edu/sekine/papers/li07.pdf> [Accedido: May. 12, 2012].
- [54]. S. AbdelRahman. "Integrated Machine Learning Techniques for Arabic Named Entity Recognition". *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 4, No 3, July 2010.
- [55]. A. Bosca, et al. "On the fly Construction of Web Services Compositions from Natural Language Requests," *Journal of Software*, vol. 1, pp. 40-50, 2006.
- [56]. M. H. t. Beek, et al. "A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods", Pisa, 2006.
- [57]. J. Sangers, "A Linguistic Approach for Semantic Web Service Discovery," Doctor in Informatics, Economics and Informatics, Erasmus University Rotterdam, Rotterdam, 2009.
- [58]. S. Angeletou, "Semantic Enrichment of Folksonomy Tagspaces", *Proc. of International Semantic Web Conference (ISWC'08)*, Doctoral Consortium. Karlsruhe, Germany, 2008.
- [59]. M. Reyes, *WATSON Exploring the Semantic Web* [Online]. United Kingdom. Knowledge Media Institute, The Open University. Disponible en: <http://kmi-web05.open.ac.uk/Overview.html>
- [60]. S. Mallick and D.S. Kushwaha, "An Efficient Web Service Discovery Architecture," *International Journal of Computer Applications*, 2010, vol. 3. p. 1-5.
- [61]. H. Qing. "Non-functional property based service selection: A survey and classification of approaches". CEUR Workshop Proceedings, 2008, 411, paper 2.
- [62]. Al-Masri and H. Mahmoud. "Discovering the best web service". Proceedings of the 16th international conference on World Wide Web (WWW'07), pp 1257-1258. 2007.
- [63]. H. Qing and H. Molina. "A modified Logic Scoring Preference method for dynamic Web services evaluation and selection," Proceedings of the 2008 IEEE Congress on Services (SERVICES'08), pp 513-520. 2008
- [64]. J.J. Dujmovic, "A Method for Evaluation and Selection of Complex Hardware and Software Systems," In Proc. of (CMG 96), vol. 1, 1996, pp. 368-378. 1996.
- [65]. P. Mika, "Ontologies are Us: a Unified Model of Social Networks and Semantics". *Web Semant*, vol. 5, pp 5-15. 2007.
- [66]. L. Specia and E. Motta "Integrating folksonomies with the semantic web," Proceedings of the 4th European conference on The Semantic Web: Research and Applications (ESWC'07). Pp 624-639. 2007.
- [67]. J.Z. Pan, S. Taylor and E. Thomas. "Enriching folksonomies Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion" 2008 [Online]: Disponible en: <http://www.abdn.ac.uk/~csc280/PTT09.pdf> [Accedido: May. 14, 2012].
- [68]. F. Nawaz, K. Qadir and H. Farooq, "SEMREG-Pro: A Semantic based Registry for Proactive Web Service Discovery using Publish Subscribe Model", *Fourth International Conference on Semantics, Knowledge and Grid*, IEEE Xplore, 2008.
- [69]. G. Wen-yue, Q. Hai-cheng and C. Hong, "Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system", *International Conference on Information Management and Engineering, IEEE Xplore*, 2010.
- [70]. G. Bond, E. Cheung, I. Fikouras, and R. Levenshteyn, "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, IPTComm '09, (New York, NY, USA), pp. 13:1–13:12, ACM, 2009.
- [71]. J. M. E. Carlin and Y. B. D. Trinugroho, "A flexible platform for provisioning telco services in web 2.0 environments," in Proceedings of the 2010 Fourth International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST '10, (Washington, DC, USA), pp. 61–66, IEEE Computer Society, 2010.
- [72]. J. C. Yelmo, et al, "User-driven service lifecycle management -adopting internet paradigms in telecom services," in Proceedings of the 5<sup>th</sup> international conference on Service-Oriented Computing, ICSOC '07, (Berlin, Heidelberg), pp. 342–352, Springer-Verlag, 2007.

- [73]. P. Falcarin, "Service composition quality evaluation in spice platform," in High Assurance Services Computing, pp. 89–102, Springer US, 2009.
- [74]. Apple. "Getting to know siri," Nov. 2011 [Online]. Disponible en: <http://media.wiley.com/productdata/excerpt/80/11182992/1118299280-40.pdf> [Accedido: Abr. 8, 2012].
- [75]. E. C. Pedraza, et al. "User-Driven Automati Resource Retrieval Based on Natural Language Request," In Proc. of (NLDB-2012) pp. 203-209. 2012.
- [76]. L.J. Suarez. "Wishful and wisdom aware composing: a full user-centric approach to create ngm," In Proc. of (WEBIST'12), pp. 803-806. 2012.
- [77]. W3C , "OWL Web Ontology Language Use Cases and Requirements ," W3C Recomendation. 2004. [Online]. Disponible en: <http://www.w3.org/TR/2004/REC-webont-req-20040210/> [Accedido: Abr. 24, 2012].
- [78]. S. Christiaens. " Metadata mechanisms: From ontology to folksonomy ... and back". *Lecture Notes in Computer Science*, pp. 199-207. 2006
- [79]. J. Morato, et al. *WordNet Applications*. In Proc. of (GWC-2004) pp. 270-278. 2004.
- [80]. Serrano, C.E., *Modelo para la Construcción de Soluciones, en Modelo Integral Para El Profesional En Ingeniería* 2005, Universidad del Cauca: Popayán. p. 75 - 94
- [81]. L. Alonso, *Herramientas libres para Procesamiento de Lenguaje Natural* [online]. Disponible en: <http://www.cs.famaf.unc.edu.ar/~laura/freeNLP> [Accedido: Jun. 10, 2012].
- [82]. D. Burquest, *Análisis Fonológico un enfoque funcional*. SIL International. 2009.
- [83]. B. Márquez, et al. *Implementación de un reconocedor de voz gratuito al sistema de ayuda a invidentes Dos-Vox en español*. Departamento de Ingeniería en Sistemas Computacionales, Universidad de las Américas Puebla. 2004.
- [84]. InCo. *Introducción al Procesamiento de Lenguaje Natural* [online]. Facultad de Ingeniería - Universidad de la República. Disponible en: <http://codigotux.wordpress.com/2009/08/05/tokenizacion/> [Accedido: Mar. 4, 2012].
- [85]. The Stanford Natural Language Processing Group. *Stanford Tokenizer* [online]. Disponible en: <http://nlp.stanford.edu/software/tokenizer.shtml> [Accedido: Ene. 12, 2012].
- [86]. GeoSoft. *Smart Tokenizer* [online]. Disponible en: <http://geosoft.no/software/tokenizer/SmartTokenizer.java.html> [Accedido: Jun. 8, 2012].
- [87]. H. Ueda, et al. *Método general de lematización con una gramática mínima y un Diccionario óptimo. Aplicación a un corpus escrito dialectal*. Universidad de Tokio, Universidad de Barcelona. 2010.
- [88]. P. Willett, *The Porter stemming algorithm: then and now*. Program: electronic library and information systems, 2006, pp. 219-223.
- [89]. M. Porter, *The Porter Stemming Algorithm* [online]. Disponible en: <http://tartarus.org/~martin/PorterStemmer/> [Accedido: May. 25, 2012].
- [90]. k. Brownridge, et al. *The Snowball Project* [online]. Agreeing metrics for research information management. Disponible en: <http://www.projectsnowball.info/wp-content/uploads/snowballproject.pdf> . [Accedido: May. 12, 2012].
- [91]. N. Haji, et al. "Porter Stemming Algorithm for Semantic Checkin," In Proc of (ICIT 2012), pp. 253-258. 2012.
- [92]. T. Lager, *Part-of-Speech Tagging: Definition*. Göteborg University. Disponible en: [http://w3.msi.vxu.se/~nivre/teaching/gslt/tagging\\_NLP1.pdf](http://w3.msi.vxu.se/~nivre/teaching/gslt/tagging_NLP1.pdf) [Accedido: Abr. 28, 2012].
- [93]. P. Cifuentes, *La semántica conceptual* [online]. Universidad Nacional Mayor de San Marcos. Disponible en: [http://www.unmsm.edu.pe/ceupseducacion/distancia\\_archivos/lbarretxe-Valenzuela-LC CAP 2.7.pdf](http://www.unmsm.edu.pe/ceupseducacion/distancia_archivos/lbarretxe-Valenzuela-LC CAP 2.7.pdf) [Accedido: Mar. 10, 2012].
- [94]. F. Ortín, *Análisis Semántico en Procesadores de Lenguaje* [online]. Lenguajes y Sistemas Informáticos - Universidad de Oviedo. Disponible en: <http://www.di.uniovi.es/procesadores/Apuntes/semantico/trans.sem.pdf> [Accedido: May. 17, 2012].
- [95]. I. Martín, *En Torno a la Polisemia y la Homonimia*. Universidad de Rioja, pp. 193- 205. 2003.
- [96]. J. Pérez, et al. *Introducción al Lenguaje de Programación Lógica Prolog*. Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica. 2007.
- [97]. S. Vázquez, *Resolución de la ambigüedad semántica mediante métodos basados en conocimiento y su aportación a tareas de PLN*. Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante. 2009.
- [98]. F. De la Caridad Fernandez, et al. "Consideraciones de diseño para una herramienta de análisis semántico". *Revista Lingüística Teórica y Aplicada (RLA 2011)*. Vol. 49 no.1, pp 51-68. 2011.
- [99]. S. Banerjee, *Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet*. Master of Science. Department of Computer Science, University of Minnesota. 2002.

- [100]. Y. Ramírez, *Métodos para el Reconocimiento de Nombres de Entidades Anidados y No Anidados en Español*, Máster en Ciencia de la Computación. Departamento de Ciencias de la Computación, Universidad de Oriente. 2008
- [101]. M. Goodchild, *Introduction to digital gazetteer research*. Department of Geography, University of California, Santa Barbara. 2008.
- [102]. GATE. *Module 3: Introduction to JAPE*. [online]. University of Sheffield NLP. 2009. Disponible en: <http://gate.ac.uk/sale/talks/gate-course-may10/track-1/module-3-jape/module-3-jape.pdf> [Accedido: Mar. 29, 2012].
- [103]. J. Véronis, et al. *Word Sense Disambiguation: The State of the Art*. Department of Computer Science, Vassar College, Poughkeepsie, New York. 1998.
- [104]. A. Schmidt and N. Skamriis, "Textual Similarity: Comparing texts in order to discover how closely they discuss the same topics," (*Kongens Lyngby 2008*) IMMBS. 2008
- [105]. G. Weber, "Tagging Technologies". *Wirtschaftsuniversität Wien Bakka Laureats Arbeit*. 0175 IT Vertiefungskurs (Deutschland) VI, pp 7-59. 2009.
- [106]. S. Hagemann, et al. "Web Service Discovery - Reality Check 2.0," *In Proc. of. (NWESP 2007)*, pp 113–118. 2007.
- [107]. A. Fernández, et al. "Closing the Service Discovery Gap by Collaborative Tagging and Clustering Techniques," *In Proc. of. (SMRR 2008)*, pp 1-13. 2008
- [108]. F. Giunchiglia, and Zaihrayeu, I. "Lightweight Ontologies," *University of Trento*. Technical Report DIT-07-071. Octubre 2007.
- [109]. P. Mika, "Ontologies are Us: a Unified Model of Social Networks and Semantics," *In Proc. of. (International Semantic Web Conference 2005)*, pp. 522-526.
- [110]. H. Halpin, V. Robu, and H. Shepard. "The dynamics and semantics of collaborative tagging". *In Proc. of. (the 1st Semantic Authoring and Annotation Workshop)*, 2006.
- [111]. C. Cattuto, et al. "Network properties of folksonomies," *In Proc. of (WWW 2007)* pp. 245–262. 2007.
- [112]. A. Passant, and P. Laublet, "Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data," *In Proc. of (LDOW2008)*, pp. 1-5. 2008.
- [113]. H. Kim, et al. "Review and alignment of tag ontologies for semantically-linked data in collaborative tagging spaces," *In Proc. of (ICSC 2008)*, pp 315–322. 2008.
- [114]. A. Mikroyannidis, "Toward a social Semantic Web.Computer," *In Proc. of (MC 2007)*, pp. 113–115, 2007.
- [115]. M. Magableh, "A Generic Architecture for Enhanced Semantic Tagging Systems," Doctor Thesis. Montfort University. United Kingdom Leicester. July 2011.
- [116]. L.J. Suarez, et al. "Service Discovery in Ubiquitous Computing Environments," *In Proc. of. (ICIW 2011)*, pp 1-9. 2011.
- [117]. H. Al-Khalifa, and H. Davis, "FolksAnnotation: A semantic metadata tool for annotating learning resources using folksonomies and domain ontologies," *In Proc. of. (Innovations in Information Technology 2006)* pp. 1-5. 2006
- [118]. F. Ghali, et al. "Folksonomies and ontologies in authoring of adaptive hipermedia," (*A3H 2008*) pp. 1-10. 2008.
- [119]. S. Bindelli, et al. "Improving search and navigation by combining ontologies and social tags," *In Proc. of. (OTM 2008 Workshops)*, pp. 76–85, 2008.
- [120]. D. Laniado, et al. "Using WordNet to turn a folksonomy into a hierarchy of concepts. *In Semantic Web Application and Perspectives,*" *In. Proc. of (SWAP 2007)*, pp. 192--201 2007.
- [121]. G. Van Heijst, et al, "Using explicit ontologies in KBS development," *International Journal of Human-Computer Studies*, 45: 183-292. 1997.
- [122]. V. Snasel, et al. "WordNet Ontology Based Model for Web Retrieval," *In Proc. of (WIRI 2005)*, pp. 200-225. 2005.
- [123]. A. Sofia, et al. "Improving search in folksonomies: a task based comparison of WordNet and ontologies," *In Proc. of (K-CAP 2009)*, pp. 169-170. 2009.
- [124]. P. Sojka, et al. "Wordnet Applications," *In Proc. of (GWC 2004)*, pp. 270-278. 2003.
- [125]. G. Miller, et al. "Introduction to WordNet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, pp. 235–244. 1990.
- [126]. N. Lin, et al. "The dynamic features of Delicious, Flickr, and YouTube," *Journal of the American Society for Information Science and Technology*. vol. 63, Issue 1, pp. 139–162, January 2012.
- [127]. U. Küster and B. Köning, "Measures for Benchmarking Semantic Web Service Matchmaking Correctness," *In Proc. of (ESWC 2010)*, pp. 45-59. 2010.
- [128]. H. Chuang, et al. 1997. Document Ranking and the Vector-Space Model, [Online]. Disponible en: <http://www.cs.ust.hk/faculty/dlee/Papers/ir/ieee-sw-rank.pdf>. [Accedido: May. 18, 2012].

- [129]. M. Montaner, et al. "A Taxonomy of Recommender Agents on the Internet," *Artif. Intell.* vol. 19. Issue 4, pp. 285–330. 2003.
- [130]. D. Tiscornia, *The Lois Project: Lexical Ontologies for Legal Information Sharing*, [Online]. Institute of Legal Information Theory and Techniques - Italian National Research Council. Disponible en: <http://www.e-p-a-p.com/dlib/9788883980466/art14.pdf> [Accedido: May. 30, 2012].
- [131]. I. Toma and D. Foxvog, "Survey of Non-Functional properties approaches," *Non-functional properties in Web services*. Deri. 2004, pp. 6-10.
- [132]. P. Eide, *Quantification and Traceability of Requirements*. Oslo, Norway. 2005.
- [133]. H. Yu and S. Reiff-Marganiec, "A Method for Automated Web Service Selection," *In Proc. of (SERVICES 2008)*, pp. 513-520. 2008.
- [134]. Su, Stanley., et al. "A cost-benefit decision model: analysis, comparison and selection of data management," *ACM Trans. Database Syst*, pp 472-520. 1997.
- [135]. Perlita labs. List of Data Visualization Tools for Designers and Developers [online]. Disponible en: <http://perlitalabs.wordpress.com/2009/11/04/list-of-data-visualization-tools-for-designers-and-developers/> [Accedido: Abr. 26, 2012].
- [136]. ASAB. "Chapter 5 Control Flow Structures," *Matlab Review Session*. 2005, pp. 70-77.
- [137]. CludFront. Introduction to Information Retrieval The Vector Space Model (VSM) [Online]. Disponible en: <https://d19vezwu8euffl6.cloudfront.net/nlp/slides%2F05-02-06-IR-VectorSpaceModel.pdf>. [Accedido: Jun. 2, 2012].
- [138]. R. Chillarege, "Software Testing Best Practices," Center for Software Engineering, IBM Research. IBM Technical Report. 1999.
- [139]. R. Mihalcea, et al. "The SENSEVAL–3 English Lexical Sample Task," *In Proc. of (The SENSEVAL-3)* 2004.
- [140]. T. Cohn, "Performance Metrics for Word Sense Disambiguation," 2003.
- [141]. M. Yatskevich, "Preliminary Evaluation of Schema Matching Systems. Department of Information and Communication Technology", University of Trento. 2003.
- [142]. B. Sun, *Named entity recognition Evaluation of Existing Systems," Master in Information Systems*. Norwegian University of Science and Technology. Norway July 2010.
- [143]. M. Marrero, "Evaluation of Named Entity Extraction Systems," *Research in Computing Science* 41, pp. 47-58. 2009.
- [144]. H. Schfitze, "Distributional Part - of Speech Tagging," *In Proc. of. (EACL 1995)*, pp. 141-148. 1995.
- [145]. OpenCloud. "Rhino performance with SIP," Rhino with SIP [Online]. Disponible en: <https://developer.opencloud.com/devportal/display/RD/Rhino+with+SIP>
- [146]. OpenCloud. "Technological evaluation," Appendix A. Evaluation Checklist [Online]. Disponible en: <https://developer.opencloud.com/devportal/display/RD/Appendix+A.+Evaluation+Checklist>
- [147]. K. Kishida, "Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments," Technical Report NII-2005-014E, National Institute of Informatics, Tokyo, Japan 2005.
- [148]. Kando et al. "Information Retrieval System Evaluation using Multi-Grade Relevance Judgments - Discussion on Averageable Single-Numbered Measures," *Joho Shori Gakkai Kenkyu Hokoku (Japanese)*, pp. 105-112, 2001.
- [149]. F. Duchateau, "Appendix: Matching and Alignment: What is the Cost of User Post-match Effort," 2009.
- [150]. M. Gawinecki, "Web Service Classification Schema," WScolab: Structured Collaborative Tagging for Web Service Matchmaking [Online]. Disponible en: <http://www.ibspan.waw.pl/~gawinec/wss/wscolab.html>
- [151]. H. Cunningham, "Jape: A JAVA Annotation Patterns Engine," *Research Memorandum CS – 99 – 06*, Department of Computer Science, University of Sheffield, may 1999.
- [152]. Rapid-i. "Rapidminer" [Online]. Disponible en: <http://rapid-i.com/content/view/181/190/> [Accedido: Abr. 7, 2012].
- [153]. NLTK. "Natural Language Toolkit" [Online]. Disponible en: <http://nltk.org/> [Accedido: Mar. 8, 2012].
- [154]. GATE, "General Architecture for Text Engineering" [Online]. Disponible en: [gate.ac.uk/](http://gate.ac.uk/) [Accedido: Ago. 10, 2011].
- [155]. Apache. "Welcome to Apache OpenNLP" [Online]. Disponible en: <http://opennlp.apache.org/> [Accedido: Mar. 12, 2012].
- [156]. WEKA, "Weka 3: Data Mining Software in Java" [Online]. Disponible en: <http://www.cs.waikato.ac.nz/ml/weka/> [Accedido: Mar. 9, 2012].
- [157]. JWebPro. "JWebPro: A Java-based Web Processing Toolkit," [Online]. Disponible en: <http://jwebpro.sourceforge.net/> [Accedido: Mar. 10, 2012].
- [158]. Apache. "Welcome to the Apache UIMA Project" [Online]. Disponible en: <http://uima.apache.org/> [Accedido: Mar. 12, 2012].

- [159]. Alias. "*LingPipe*," [Online]. Disponible en: <http://alias-i.com/lingpipe/>
- [160]. Orange. "*NLP Trainers Council*" [Online]. Disponible en: <http://www1.nlp.ru/eng/indnlp.php3>
- [161]. B. Plannerer. "Chapter 1 - The Speech Signal," *An Introduction to Speech Recognition*. Plannerer, B. Munich, Germany. 2005, pp 1-16.
- [162]. C. L. Dominguez, "*Lingüística: una introducción generalísima*". Grupo de Lingüística Hispánica (GLH) del Departamento de Lingüística de la Universidad de Los Andes. Mérida-Venezuela. 2002.