

CÓDIGO: CIS0830IS02

Sistema de Enriquecimiento de consultas basado en los perfiles de usuario y de contexto tomando en cuenta el perfil de sesión y el histórico en el sistema

<http://pegasus.javeriana.edu.co/~CIS0830IS02/>

FERNANDO ANTONIO ARAGON MANJARRES

MARÍA CLAUDIA HIGUERA PALACIO

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA DE SISTEMAS

BOGOTÁ, D.C.

2009

CÓDIGO: CIS0830IS02

Sistema de Enriquecimiento de consultas basado en los perfiles de usuario y de contexto
tomando en cuenta el perfil de sesión y el histórico en el sistema

<http://pegasus.javeriana.edu.co/~CIS0830IS02/>

Autores:

Fernando Antonio Aragón Manjarrés

María Claudia Higuera Palacio

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS
REQUISITOS PARA OPTAR AL TITULO DE INGENIERO DE SISTEMAS

Director

Ángela Cristina Carillo Ramos

Julio Carreño

Jurados del Trabajo de Grado

María Consuelo Franky de Toro

Enrique González Guerrero

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA DE SISTEMAS

BOGOTÁ, D.C.

DICIEMBRE, 2009

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS

Rector Magnífico

Joaquín Emilio Sánchez García S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Francisco Javier Rebolledo Muñoz

Decano del Medio Universitario Facultad de Ingeniería

Padre Sergio Bernal Restrepo S.J.

Director de la Carrera de Ingeniería de Sistemas

Ingeniero Luis Carlos Díaz Chaparro

Director Departamento de Ingeniería de Sistemas

Ingeniero Germán Alberto Chavarro Flórez

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

Tabla de contenido

1. INTRODUCCIÓN	1
2. DESCRIPCIÓN GENERAL DEL TRABAJO DE GRADO	2
2.1. PROBLEMÁTICA.....	2
2.2. OBJETIVOS.....	4
2.2.1. <i>Objetivo General</i>	4
2.2.2. <i>Objetivos específicos</i>	5
3. ESTADO DEL ARTE.....	6
3.1. ADAPTACIÓN	6
3.1.1. <i>Tipos de adaptación</i>	8
3.1.1.1. Perfil contextual	8
3.1.1.1.1. Componentes contextuales	9
3.1.1.1.2. Trabajos relacionados.....	10
3.1.1.2. Perfil de usuario	11
3.1.1.2.1. Componentes del perfil de usuario	12
3.1.1.2.2. Trabajos relacionados.....	14
3.1.1.3. Formas de representación de los perfiles	14
3.1.1.4. Adquisición de información para los perfiles	16
3.1.1.5. Priorización de componentes.....	16
3.2. ENRUTAMIENTO DE CONSULTAS	18
3.2.1. <i>Actividades que involucra el proceso</i>	18
3.2.2. <i>Análisis de consultas</i>	19
3.2.3. <i>Reescritura de consultas</i>	21
3.2.3.1. Consultas SQL.....	21
3.2.3.1.1. Operadores Clausula <i>WHERE</i>	21
3.2.3.2. Trabajos relacionados con la reescritura de consultas.....	22
3.2.4. <i>Selección de fuentes de información</i>	23
3.2.5. <i>Creación de cadenas en un buscador</i>	25
3.2.6. <i>Recepción de resultados</i>	26
3.3. ENRIQUECIMIENTO DE CONSULTAS	27
3.3.1. <i>Actividades que pertenecen al proceso</i>	28
3.3.2. <i>Trabajos relacionados</i>	29
4. METODOLOGÍA.....	30
4.1. METODOLOGÍA PROPUESTA	30
4.2. METODOLOGÍA REALIZADA.....	31
5. CONTRIBUCIONES.....	31
5.1. SISTEMA DE ENRIQUECIMIENTO DE CONSULTAS	32
5.2. ESQUEMATIZACIÓN DEL PROCESO DE ENRIQUECIMIENTO.....	33
5.3. MÓDULO DE ENRIQUECIMIENTO DE CONSULTAS	34
5.3.1. <i>Definición del perfil de usuario</i>	36
5.3.1.1. Componentes del perfil de usuario	37

5.3.1.2.	Representación de los elementos del perfil de usuario	38
5.3.1.3.	Comunicación entre los componentes del perfil	39
5.3.2.	<i>Perfil contextual</i>	40
5.3.2.1.	Componentes del perfil contextual	41
5.3.2.2.	Representación de los elementos del perfil de contexto	43
5.3.2.3.	Comunicación entre los componentes del perfil contextual	44
5.3.3.	<i>Comunicación entre el perfil de usuario y el perfil de contexto</i>	44
5.3.4.	<i>Cambios a los perfiles</i>	46
5.3.5.	<i>Reglas</i>	47
5.3.5.1.	Reglas de equivalencia.....	48
5.3.5.2.	Reglas de componente	51
5.3.5.3.	Persistencia de las reglas	54
5.3.5.3.1.	Persistencia de las reglas de equivalencia.....	54
5.3.5.3.2.	Persistencia de las reglas de componente	55
5.3.6.	<i>Priorización de las reglas</i>	56
5.3.6.1.	Priorización de componentes.....	57
5.3.6.2.	Priorización de reglas de componente.....	58
5.3.7.	<i>Perfil de sesión</i>	58
5.3.8.	<i>Perfil histórico</i>	60
5.3.9.	<i>Reescritura de consultas</i>	61
5.3.10.	<i>Dinamismo del contexto</i>	64
5.3.10.1.	Generalización del proceso	65
5.3.10.2.	Componentes del registro de cambio.....	66
5.3.10.3.	Conflictos.....	67
5.3.10.4.	Priorización de componentes.....	68
5.3.10.5.	Margen de tolerancia	68
6.	CASO DE ESTUDIO	69
6.1.	ARQUITECTURA DEL PROTOTIPO	70
6.2.	CASOS DE USO	73
6.3.	PRUEBAS FUNCIONALES	74
6.4.	PRUEBAS UNITARIAS Y DE INTEGRACIÓN.....	74
6.5.	TIPOS DE PRUEBA.....	75
6.6.	RESULTADOS	75
6.7.	PRUEBAS DE USUARIO.....	77
7.	CONCLUSIONES Y TRABAJOS FUTUROS	78
7.1.	CONCLUSIONES	78
7.2.	TRABAJOS FUTUROS	79
8.	REFERENCIA	81
9.	ANEXOS	89

Lista de figuras

Figura 1. Arquitectura del sistema de enriquecimiento.....	5
Figura 2. Componentes Bucur et al. [17]	10
Figura 3. Perfil de usuario propuesto por MAIPU.....	13
Figura 4. Arquitectura de <i>MAICO</i>	13
Figura 5. Representación de contexto, tomado de [42].....	15
Figura 6. Perfil de preferencias con prioridades establecidas.....	17
Figura 7. Priorización de componentes tomado de [26].....	17
Figura 8. Análisis de consulta [29]	20
Figura 9. Enrutamiento de consultas, tomado de [38].....	24
Figura 10. Selección y clasificación de pares [72].....	25
Figura 11. Sistema de filtrado, adaptado de[61]	26
Figura 12. Proceso de búsquedas Web [69]	29
Figura 13. Metodología de la propuesta del trabajo de grado	30
Figura 14. Metodología realizada	31
Figura 15. Sistema de enriquecimiento de consultas	32
Figura 16. Diagrama del proceso de enriquecimiento de consultas.....	33
Figura 17. Funcionamiento del Módulo de Enriquecimiento de Consulta y generación de la consulta enriquecida a partir del perfil de sesión y la consulta original.	35
Figura 18. Proceso de desarrollo del sistema de enriquecimiento.	35
Figura 19. Perfil contextual dentro de la proceso	36
Figura 20. Perfil de usuario propuesto	37
Figura 21. Ontología del perfil de usuario.....	39
Figura 22. Comunicación entre componentes del perfil de usuario.....	39
Figura 23. Perfil contextual dentro de la proceso	40
Figura 24. Perfil contextual utilizado en el módulo de enriquecimiento.....	41
Figura 25. Representación ontológica del perfil de contexto	44
Figura 26. Interacción entre componentes contextuales	44
Figura 27. Proceso para realizar cambios en los perfiles	47
Figura 28. Definición de reglas dentro del proceso	47
Figura 29. Definición de reglas	48
Figura 30. Generalización de las reglas de equivalencia.....	50
Figura 31. Esquema de adición de términos.....	50
Figura 32. Ejemplo de adición de términos.....	50
Figura 33. Esquema de reemplazo de términos	51
Figura 34. Ejemplo de reemplazo de términos	51
Figura 35. Esquema de exclusión de términos	51
Figura 36. Ejemplo de exclusión de términos	51
Figura 37. Ejemplo de regla valor-componente (sencillas).....	52
Figura 38. Ejemplo de regla compuesta de valor-componente.....	53
Figura 39. Ejemplo de regla de componente-componente (compleja) e instanciación	53
Figura 40. Ejemplo de regla de componente comparando valores de componentes.....	53

Figura 41. Ejemplo de regla de componente respecto a rol.....	53
Figura 42. Esquema de una relación de equivalencia	54
Figura 43. Esquema de condición para reglas de equivalencia	54
Figura 44. Esquema de escritura de adición	55
Figura 45. Esquema de escritura de reemplazo	55
Figura 46. Esquema de escritura de Exclusión	55
Figura 47. XML con reglas de equivalencia	55
Figura 48. Especificación de componente a comparar y de comparación.....	56
Figura 49. Esquema de relación entre componentes	56
Figura 50. XML de reglas de componente	56
Figura 51. La priorización de las reglas dentro del proceso	56
Figura 52. Asignación de pesos a los componentes	58
Figura 53. El perfil de sesión dentro del proceso.....	58
Figura 54. Representación del perfil de sesión	59
Figura 55. Perfil histórico dentro del proceso	60
Figura 56. Representación del perfil histórico	60
Figura 57. La reescritura de consultas dentro del proceso	61
Figura 58. Porcentajes de ingreso a los motores de búsqueda [4]	63
Figura 59. El dinamismo del contexto dentro del proceso	64
Figura 60. Proceso de configuración del registro de cambios	65
Figura 61. Proceso de uso del registro de cambios	65
Figura 62. Ontología del registro de cambios.....	66
Figura 63. Arquitectura del prototipo <i>VIVASUM</i>	70
Figura 64. Diagrama entidad relación del módulo de enriquecimiento	72
Figura 65. Diagrama Entidad Relación de la aplicación <i>VIVASUM</i>	72
Figura 66 Pruebas por casos de uso	75
Figura 67 Estado de los casos de prueba.....	76
Figura 68 Estadística tipos de prueba.....	76
Figura 69 Cantidad de bugs	77

Lista de tablas

Tabla 1. Comparación de artículos de contexto.....	11
Tabla 2. Comparación trabajos perfil de usuario	14
Tabla 3. Operadores de comparación, tomado de [59]	22
Tabla 4. Comparación de trabajos de reescritura de consultas	23
Tabla 5. Comparación trabajos relacionados enriquecimiento.	29
Tabla 6. Perfil de usuario como proveedor, perfil contextual como cliente	45
Tabla 7. Perfil contextual como proveedor, perfil de usuario como cliente	45
Tabla 8. Relaciones de equivalencia	49
Tabla 9. Posibles escenarios para las relaciones de equivalencia	50
Tabla 10. Relaciones de componente sencillas.....	52
Tabla 11. Ejemplo relaciones de componente complejas.....	52
Tabla 12. Adaptado de [36]	63
Tabla 13. Reescritura a partir de las reglas.....	64
Tabla 14. Especificación de rangos de tolerancia	68
Tabla 15. Características no probadas.....	77

Abstract

Information Systems (*SI*) are getting more robust each day and the available data amount grows constantly. Sometimes, information does not correspond to users' needs and however, it is shown to them when a query is made, creating a cognitive overcharge. In order to reduce the amount of futile information presented, content adaptation can be used to give users information according to their needs. In order for this adaptation to take place, user's characteristics and likes, as well as information about the context surrounding the interaction between the user and the system have to be taken into account. This work presents a system that aims to adapting user given queries, adapting them to that particular user's needs.

Resumen

Los Sistemas de Información (*SI*) son cada vez más robustos y la cantidad de datos disponible crece constantemente. Algunas veces, la información no corresponde a las necesidades de los usuarios y sin embargo, se les presenta cuando se hace una consulta, lo que crea una sobrecarga cognitiva. Para reducir la cantidad de información inoficiosa¹ entregada, se puede recurrir a la adaptación al contenido, la cual permite entregar al usuario información acorde a sus gustos, preferencias y contexto de uso, es decir a las necesidades del usuario. Para realizar dicha adaptación, es necesario tener en cuenta las características y los gustos del usuario, así como información del contexto en el cual el usuario interactúa con el sistema. El presente trabajo presenta un sistema de enriquecimientos que se encarga de aumentar las consultas entregadas por un usuario, con el fin de adaptarla según sus las características de la interacción con el sistema, sus gustos y preferencias.

¹ Información que no corresponde a las necesidades del usuario

Resumen ejecutivo

Cuando un usuario necesita información genera una consulta, que es enviada a diversos sistemas de información (SI) con el fin de contestarla. Algunos de estos SI manejan un gran volumen de información y constantemente se añaden más, con diferentes niveles de calidad; esta heterogeneidad puede causar la entrega de resultados no acordes a las necesidades del usuario. A partir de lo descrito anteriormente, se pueden presentar los siguientes problemas: *i) la desorientación*, es decir el no ver claridad caminos a seguir con respecto a la interacción con la información; *ii) la sobrecarga cognitiva*, que se presenta cuando los usuarios reciben un gran volumen de información y no saben cómo procesarla; y por último, *iii) la ausencia de personalización* del contenido, lo cual quiere decir que no se tiene en cuenta quien es el usuario, sus características y el contexto en el que interactúa con el sistema, y que por esto se podría entregar información que no satisfaga las necesidades del usuario[54]. Para evitar los problemas descritos anteriormente y para lograr que los resultados presentados se ajusten más a las necesidades de los usuarios, se deben tener en cuenta sus características particulares, es decir, quién es el usuario, su rol dentro de la comunidad, sus gustos, preferencias e información sobre cómo y cuándo interactúa con el sistema. Para esto se definió un sistema de enriquecimiento de consultas a partir de un perfil del usuario y un perfil del contexto.

En primera instancia, se define un perfil de usuario y un perfil contextual, que se encargan de modelar tanto al usuario, como las características de la interacción del mismo con el sistema. Para el perfil de usuario se consideran características como los datos básicos, los gustos, las preferencias y el rol del usuario dentro de una comunidad. Por su parte, el perfil de contexto contempla la localización del usuario, las características estructurales del sitio donde se encuentra, las actividades que está realizando o que realizara en un futuro y las características ambientales (por ejemplo la cantidad de luz y ruido).

A partir de los datos consignados en el perfil de usuario y en el de contexto, se procede a realizar el enriquecimiento a la consulta suministrada por el usuario, es decir, se reescribe la consulta. Para realizar dicha reescritura, se establecieron dos tipos de reglas: *i) las reglas de equivalencia* y *ii) las reglas de componente*. Las reglas de equivalencia que establecen una especie de diccionario de sinónimos y las reglas de componentes, que definen que modificaciones hacer a una consulta, partiendo de los datos consignados en los perfiles (*e.g.* si la actividad actual es trabajo, se agrega a la consulta “java” el concepto “programación”).

Luego de enriquecer la consulta con los datos contextuales y de usuario, se debe tener en cuenta los cambios que se puedan presentar dentro del contexto, antes de entregar los resultados al usuario. Para esto, se consideró el uso de un registro de cambios, por medio del cual se establece si es necesario volver a realizar una reescritura de consultas para entregar datos acordes a las necesidades del usuario, utilizando márgenes de tolerancia. Los márgenes de tolerancia establecen el rango fuera del cual un cambio contextual debe ser tenido en cuenta, por ejemplo, si se establece un margen de tolerancia de un día, y la entrega de resultados se demora más que esto, se deberá reescribir la consulta.

Para finalizar, se realizó un prototipo llamado Viajes y Vacaciones a SU Medida (*VIVASUM*), con el fin de probar las contribuciones presentadas. *VIVASUM* es una agencia de viajes que tiene como fin prestar servicios personalizados, al utilizar el sistema de enriquecimiento de consultas propuesto en el presente trabajo de grado.

1. Introducción

Cuando un usuario genera consultas a uno o más Sistemas de Información (*SI*), podría recibir información que no se ajuste a sus necesidades o a las características en torno a su interacción con dichos sistemas. Para propósitos de adaptación, cuyo fin último es brindarle a un usuario la información apropiada, en un momento y lugar determinado, las consultas deberían incluir parámetros como las restricciones de despliegue o de contenido que establezca el usuario, e incorporarlas en la consulta para enviarlas a los *SI* que las puedan satisfacer.

La cantidad de información y así mismo, de sistemas o fuentes de información crece constantemente [81]. La información que será mostrada a los usuarios, puede provenir de múltiples fuentes que pueden no ser las más apropiadas para responder a una consulta específica. Por tal motivo, la decisión de qué *SI* puede satisfacer dicha consulta es tarea de los mecanismos tradicionales de enrutamiento [59][67]. Para esto, se utilizan métodos tales como confianza y reputación [3], agrupación de pares [43] o correspondencia de datos [44].

Además de la gran cantidad de información, ésta (la información), es cada vez es más compleja semántica y sintácticamente², con lo cual se pueden presentar problemas de concordancia entre los criterios buscados y los resultados obtenidos. Los criterios de selección de información sencillos, tales como la comparación de cadenas de caracteres (sin tener en cuenta el significado semántico en el contexto de la consulta), tienden a ser cada vez menos eficientes y precisos en cuanto a los resultados presentados al usuario, lo cual quiere decir, que se presenta cada vez menos información que se ajuste a las preferencias, características y necesidades de los usuarios. Tampoco son tenidas en cuenta las características contextuales, es decir, aquellas que describen el entorno en el cual se lleva a cabo la interacción entre el usuario y el sistema. La selección de información es generalizada, lo cual quiere decir que, se presenta la misma información a todos los usuarios. La situación ideal atañe entonces, a la necesidad de incorporar en las consultas las preferencias de usuario y las características contextuales, con el fin de proveer al usuario los resultados que satisfagan sus necesidades específicas, en un momento determinado.

² La complejidad semántica se refiere la información que contiene palabras con el mismo significado y a la que contiene homónimos. Por su parte, la complejidad sintáctica hace referencia a los conectores lógicos utilizados en las búsquedas.

A continuación se presentará la problemática y los objetivos del presente trabajo, luego se presenta el estado del arte, donde se contemplan temas como la *i)* adaptación para poder modificar las consultas que ingresa un usuario, para que sean acordes con las necesidades de usuario; *ii)* el enrutamiento de consultas, que indica el proceso que se lleva a cabo para que una consulta obtenga respuesta(s) por parte de alguna(s) fuente(s) de información y *iii)* el enriquecimiento de consultas, es decir el proceso que se lleva a cabo para adaptar una consulta inicial, dada por el usuario con el fin de obtener resultados acordes para dicho usuario. A partir de la información recolectada en el estado del arte, se procede a mostrar la metodología utilizada para plantear las contribuciones, a partir de las cuales se determinan las conclusiones y los trabajos futuros.

2. Descripción general del trabajo de grado

En el presente capítulo se muestra la problemática en la que se enmarca el desarrollo del presente trabajo de grado, junto con la pregunta motivadora, para luego se describir los objetivos.

2.1. Problemática

La diversificación de las necesidades del usuario ha generado el surgimiento de nuevos retos y a su vez de nuevas tecnologías como la computación móvil, la computación ubicua y *SI* cada vez más robustos, con mayor cantidad de información y con altas capacidades de interacción con los usuarios; esto supone nuevos retos y campos de acción para satisfacer dichas necesidades.

Cada vez que un usuario necesita información, realiza una consulta que puede remitirse a uno o más *SI*, con lo que se deben tener en cuenta diversos factores a la hora de obtener los resultados esperados. Entre estos factores se puede pensar en la heterogeneidad de la estructura de los *SI*, en las diversas modalidades de acceso a éstos y la complejidad de obtener resultados, generada por la cantidad de fuentes de información de las cuales provienen los resultados [20].

Es por esto mismo, que los resultados de la consulta del usuario, no siempre son los deseados con respecto a su contexto (conjunto de características tales como lugar, condiciones climáticas, necesidades de información, entre otras)³, ni a sus preferencias o gustos.

En cuanto al contenido, si un usuario desea buscar información acerca de la palabra “*JAVA*”, por ejemplo en un motor de búsqueda del estilo “*Google*”, normalmente obtendría resultados sobre un lenguaje de programación, lo que podría no ser útil si el usuario no es informático. Si

³ En este trabajo, contexto y contexto de uso son sinónimos.

la persona que hace la consulta está interesada en temas de geografía o desea obtener información sobre sitios para sus próximas vacaciones, los resultados deberían corresponder a información sobre la isla de Java (archipiélago indonesio); asimismo, si el usuario es un catador de café, le interesaría obtener como resultado la historia, origen y características de la variedad de café Java. La situación en la cual un usuario obtiene información que no se ajusta a su situación actual, puede obedecer al envío directo de las consultas a las fuentes de información⁴, sin antes tener en cuenta aspectos que podrían adaptar los resultados de dicha consulta a sus verdaderas necesidades. Entre dichos aspectos se pueden mencionar las preferencias y gustos del usuario y el contexto de uso⁵. En el caso en el que el sistema tome en cuenta aspectos de adaptación, éste adquiere el nombre de “sistema adaptativo”.

El fin del enriquecimiento de consultas es aumentar la concordancia entre lo que el usuario necesita y lo que obtiene, eliminando posibles sobrecargas cognitivas con temas no relacionados a sus intereses, ni con el contexto en el que se encuentra. Se podría disminuir también la cantidad de resultados entregados al usuario, lo que haría más eficaz el hallazgo de lo que él necesita entre los resultados obtenidos; además, se espera que se reduzcan las apariciones de datos redundantes y en ocasiones innecesarios.

Debido a que se desea entregar resultados acordes al contexto del usuario, es necesario considera un registro de cambios, en el que se reflejan las variaciones que pueden existir en el contexto, desde el momento en que se realiza la consulta hasta el momento en el que se le entregan los resultados al usuario. Con este registro de cambios se le da al usuario y al sistema la posibilidad de contemplar el dinamismo del contexto.

Además del registro de cambios se considera un perfil histórico y uno de sesión, lo cual indica que la información histórica de lo que el usuario quiere y necesita es importante para generar consultas más precisas; lo que evitaría el hecho de que el usuario deba ingresar para cada sesión información que ya ha ingresado al sistema, y que pueda obtenerse de sesiones anteriores.

⁴ De ahora en adelante se tratará en esta propuesta de manera indiferente el término fuente o sistema de información. En general se manejará el término Fuente de Información (*FI*).

⁵ En esta propuesta, contexto de uso y contexto son sinónimos. Este término se refiere a los elementos de información (físicos, sociales, interés del usuario, características del *DA*, entre otros) que consideras pertinentes para la interacción del usuario con una aplicación [30]

El enriquecimiento de consulta puede ser ajustado a cualquier tipo de sistema (*e.g.*, motores de búsqueda, acceso Web o sistemas convencionales). La diferencia en el mecanismo de enriquecimiento⁶ depende de la manera en la que cada sistema construye la consulta, esto puede darse desde la construcción de la cadena de los términos necesarios para la consulta en un motor de búsqueda, hasta la modificación de la cláusula “*WHERE*” de una consulta construida en *SQL*.

Después de haberse expuesto la problemática, y mostrando las fortalezas que tiene tanto la adaptación como el enriquecimiento de consultas para un adecuado tratamiento de las mismas, surge la siguiente inquietud:

¿Cómo generar un sistema que enriquezca las consultas de un usuario, basándose tanto en las preferencias y características de este último consignadas en su perfil, como en las condiciones del entorno de su interacción con los SI tomando en cuenta la información histórica así como la correspondiente a su sesión actual?

2.2.Objetivos

Partiendo de la problemática anteriormente descrita, surgen los siguientes objetivos dentro del trabajo de grado.

2.2.1.Objetivo General

Construir un sistema de enriquecimiento de consultas que, basado en las características del perfil de usuario y de su contexto, le permita al usuario obtener los resultados de dichas consultas adaptados a sus necesidades de información.

⁶ Aunque el enriquecimiento está dirigido a cualquier tipo de sistema, para efectos de este trabajo se planteará un caso de estudio dirigido a motores de búsqueda (al estilo Google) ó sistemas que utilicen lenguajes de consulta al estilo *SQL*. Esto se definirá en la actividad correspondiente al diseño y planteamiento del caso de estudio.

2.2.2. Objetivos específicos ⁷

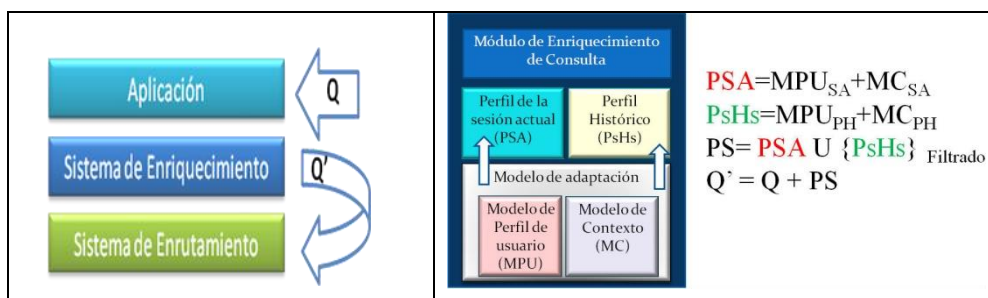


Figura 1. Arquitectura del sistema de enriquecimiento

1. Generar un modelo de adaptación que incluya (ver Figura 1):
 - Un modelo de perfil de usuario (*MPU*) basado en sus preferencias y características básicas con el fin de personalizar la información.
 - Un modelo de *contexto de uso* (*MC*) que incluya características que puedan afectar o modificar las preferencias y/o necesidades de información del usuario (*e.g.*, lugar, clima, hora del día, día de semana, fecha, días festivos, entre otros).
2. Seleccionar los componentes del modelo de adaptación que pueden ser aplicables a la sesión actual (*i.e.*, generar el *PSA*), al igual que aquellos que pueden ser extraídos del histórico de las interacciones de usuario con el sistema y que después de filtrados, no entren en conflicto con el perfil de la sesión actual (*i.e.*, generar el *PsHs*).
3. Diseñar un módulo de enriquecimiento de consultas que dada una consulta *Q* proveniente de una aplicación, genere una consulta enriquecida *Q'* que abarca los elementos contextuales y de preferencias de usuario, tanto de la sesión actual como aquellos del histórico que no entran en conflicto, para luego dirigirla (*Q'*) a un sistema de enrutamiento que la tome y la redirija a los *SI* que puedan dar una respuesta total o parcial a *Q'* (el sistema de enrutamiento se encargará del tratamiento de consultas).
4. Construir una sentencia que exprese para *Q* la inclusión de componentes de los perfiles de usuario y de contexto con el fin de generar *Q'* (por ejemplo, afectar la cadena que aparece en el *WHERE* de lenguajes como SQL sin tomar en cuenta operadores como *order by*, *group by* o la cadena que se ingresa en motores de búsqueda).

⁷ Los objetivos del 1-3 serán tratados durante TG1 y los restantes (4-6) en TG2.

5. Generar un método tanto para la generación del registro de cambios de las características contextuales (al interior del sistema de enriquecimiento de consultas), como para la variación de los términos de la consulta ante el dinamismo del contexto de uso (ver Figura 1).
6. Construir un prototipo funcional que utilice los servicios del sistema de enriquecimiento de consultas con el fin de obtener sus resultados, adaptados a las necesidades del usuario.

3. Estado del arte

El presente capítulo está organizado de la siguiente manera: en la primera parte se presenta un acercamiento a la adaptación, los tipos de adaptación existentes y los criterios que se deben tener en cuenta para realizar este proceso; luego se introduce al tema de perfil de usuario y perfil contextual, utilizados para la adaptación con respecto al contenido. Continuando así, con la descripción del enrutamiento de consulta, los métodos utilizados para hacer enrutamiento y la forma en que son tratadas las consultas; todo esto con el fin de redirigir las consultas a los sistemas de información (*SI*) que puedan responderlas total o parcialmente. Para finalizar, se describe el enriquecimiento de consultas, tema central del presente trabajo de grado.

3.1. Adaptación

Cuando un usuario busca información, ésta normalmente es entregada de forma generalizada, es decir, no se tiene en cuenta ni la heterogeneidad de los usuarios, ni las características de la interacción de los mismos con el sistema. Al no tener en cuenta estos puntos, el usuario termina recibiendo información que no corresponde a sus características y necesidades particulares, lo que podría no ser útil, tanto por su presentación, como por su contenido, en gran medida, debido a que a la cantidad de información que se encuentra en los sistemas de información (*SI*) aumenta constantemente. Ésta es heterogénea y de diversa calidad[8], y a su vez, los usuarios proporcionan consultas incompletas, es decir, en la mayoría de los casos, las consultas no proveen toda la información necesaria para entregar resultados acordes a las necesidades del usuario. Tanto así, que algunas veces un usuario espera resultados diferentes al introducir la misma consulta. Por ejemplo, un usuario busca la palabra “java”, con el deseo de obtener resultados sobre el lenguaje de programación y luego, este mismo usuario introduce la palabra “java”, pero ahora con el deseo de obtener información de la variedad de café [12][84]. En este caso, se podría considerar si el usuario está trabajando, caso en el cual podría estar interesado en “java” como un lenguaje de programación, o si por el contrario se encuentra fuera de las horas laborales podría interesarse en el café.

La adaptación busca entonces, entregar al usuario información que se ajuste a sus necesidades en un momento determinado. En el caso del ejemplo anterior, la adaptación buscaría definir la palabra “java” con diferentes significados, dependiendo de la actividad que realice el usuario. A partir de dicha adaptación, el usuario podría sentir que el sistema está hecho para él⁸, y es por esto mismo que el poder brindar a los usuarios información adaptada, se ha convertido en una prioridad para las ciencias tecnológicas[70].

Murray *et al.*[54] describen algunos problemas a los que el usuario se ve expuesto al ingresar a los SI: *i)* la *desorientación*, que no permite ver con claridad caminos a seguir con respecto a consultas sobre la información o interacción con ella; *ii)* la *sobrecarga cognitiva*, se presenta cuando los usuarios reciben un gran volumen de información y no saben cómo procesarla; y por último, *iii)* la *ausencia de personalización* del contenido, en donde no se tiene en cuenta quién es el usuario, su nivel de conocimiento y ni las aptitudes que posee; por lo cual, éste puede aburrirse con la información obtenida ya sea por su facilidad o por su complejidad.

Del análisis de los tres problemas anteriores, puede verse la adaptación como una de las posibles soluciones, debido a que es definida como la posibilidad de ajustar la información de acuerdo con las necesidades de los usuarios, sus capacidades y el contexto donde se desenvuelven e interactúan con diversos SI [10]. Por ejemplo, para resolver el problema de la *desorientación*, se puede adaptar la navegación que debe hacer el usuario en un sitio web, lo cual consistiría en proponer una ruta guiada[51], donde al usuario se le presenta la información poco a poco, dependiendo de sus necesidades y preferencias es decir accediendo progresivamente⁹ a la información[74], con lo cual se disminuiría la *sobrecarga cognitiva* a la que se puede ver expuesto el usuario. Al adaptar el contenido, dependiendo de las características de un grupo de usuarios, como el nivel educativo, gustos, formatos aceptados por el dispositivo de acceso, entre otras, el usuario podría obtener la información que más se ajuste a sus características. Partiendo de la adaptación, se pueden considerar los gustos de un único usuario, por medio de la personalización (caso particular de la adaptación), en donde se tienen en cuenta las preferencias de un usuario, para que se vean reflejadas en los resultados obtenidos [30].

⁸ Tal es el caso de un reforzador conductual positivo, debido a que el usuario siente que sus gustos son tenidos en cuenta por un sistema, al no se entrega información generalizada, sino personalizada. Lo cual, sería el refuerzo de la conducta de recurrir al sistema. [68]

⁹ Esta idea hace referencia al concepto de “acceso progresivo” expuesto por Villanova [74] que consiste en brindarle al usuario “poco a poco” la información que éste necesita y que realmente le sea útil en el momento en que la accede.

3.1.1. Tipos de adaptación

La adaptación puede ser clasificada dependiendo de la información tratada. Por una parte se encuentra la adaptación al *contenido*, que indica los ajustes que se aplicarán a los datos que se entregan al usuario. Por otra parte, se cuenta con la adaptación a la forma en la que se presenta la información, también conocida como la adaptación a la *navegación*[16]. A continuación se definirá la adaptación al contenido, la cual será base para adaptar las consultas del usuario en el presente trabajo.

La adaptación al contenido es realizada sobre la información que será entregada al usuario; para realizar dicha adaptación se han definido varias opciones, entre las que se encuentra el modelar al usuario y su contexto¹⁰ a través de un perfil de usuario y un perfil contextual respectivamente. Al respecto, Jameson [37] hace hincapié en la necesidad de modelar tanto el usuario como las características contextuales, con el fin de modelar tanto los gustos y preferencias de los usuarios¹¹, como los elementos que pueden ser utilizados para representar la interacción entre el usuario y el sistema, buscando así adaptar la información de acuerdo a las necesidades de los usuarios.

A continuación se definirá la forma en que se modela el contexto de uso, con sus respectivos componentes, para luego explicar el perfil de usuario, aspectos fundamentales considerados para la adaptación al contenido.

3.1.1.1. Perfil contextual

La adaptación busca, entregar al usuario la información adecuada respecto a sus necesidades. Para considerar las características del entorno se incluye lo que se llamará en este trabajo “*contexto*”.

Dey *et al.* [26] definen el contexto como un conjunto de elementos que pueden ser utilizados para representar la interacción entre el usuario y el sistema. Para esta definición se cuenta con tres tipos de características: *i*) físicas (*e.g.*, la localización, y tiempo), *ii*) sociales (*e.g.*, objetos y personas en el entorno del usuario), y *iii*) emocionales (*e.g.*, definido como los intereses del usuario).

¹⁰ También conocido como “Contexto de Uso”

¹¹ Los gustos se refieren a una preferencia que no cambian a largo plazo, mientras que las preferencias de los usuarios indican predilecciones que varían a menudo. [57]

También existen aplicaciones que dependen del contexto de uso y son denominadas *sensibles al contexto* [26]. El perfil contextual busca brindar los componentes necesarios para un correcto funcionamiento de éstas aplicaciones. Por ejemplo, si una aplicación necesita la ubicación de la persona y la actividad que se encuentra realizando, entonces el perfil contextual debe incluir dichos componentes, para que la aplicación pueda hacer uso de ellos.

A su vez, Bucur *et al.*[17] definen como principal problemática de las aplicaciones sensibles al contexto y del contexto en sí, la gran cantidad de información contextual que se puede presentar. Es por esto, que se debe definir un perfil contextual, para escoger la información que es necesario tener en cuenta.

A continuación se presentarán los componentes que diversos autores incluyen dentro del perfil contextual.

3.1.1.1.1. Componentes contextuales

El perfil contextual puede estar conformado por muchos elementos, y estos a su vez pueden cambiar dependiendo de los autores que desarrollen el perfil y de las aplicaciones que lo utilicen. Incluso se puede llegar a considerar al usuario mismo como parte del contexto.

Por ejemplo, Dey *et al.* [30] proponen una clasificación de los componentes generales que debería contemplar una representación del contexto de uso. Se considera que el contexto es un conjunto de características físicas, sociales y emocionales; las primeras contemplan la localización del usuario, la fecha actual y el tiempo de conexión, entre otros aspectos. Las segundas tratan sobre las situaciones sociales que rodean al usuario, es decir, modela los objetos y las personas que rodean al usuario. Y las características emocionales son los intereses de los usuarios.

Por su parte Bucur *et al.* [17] definen dentro del contexto diferentes componentes. Cada uno de estos consta de un nombre y una lista de atributos relacionados (Ver Figura 2). El perfil contextual incluye las características de una persona, el tiempo, la localización, las actividades, una agenda, los dispositivos y tecnologías disponibles.

Person – related InterestsPerson : (Person) -> String IsSupervisorOf : (Person, Person) -> Boolean StatusPerson : (Person) -> String Supervises : (Person) -> Person* RoleOfPersonInGroup : (Person, Group) -> Role	Time-related TimeZone : (Time) -> Integer DayOfWeek : (Date) -> String TimeOfDay : (Time) -> String
Location - related PersonsInRoom : (Person, Room) -> Boolean PersonsAtFloor : (Person, Floor) -> Boolean PersonsInBuilding : (Person, Building) -> Boolean PersonsInCity : (Person, City) -> Boolean	Activity – related ActivityStartsAt : (Activity) -> Time ActivityEndsAt : (Activity) -> Time ActivityGoal : (Activity) -> String ActivityDuration : (Activity) -> Integer ActivityParticipants : (Activity) -> Person*
Agenda - related BusyMorning : (Agenda) -> Boolean BusyAfternoon : (Agenda) -> Boolean BusyEvening : (Agenda) -> Boolean	Environment – related DevicesAvailableInBuilding : (Building) -> Device* DevicesAvailableInRoom : (Room) -> Device* DevicesAvailableAtFloor : (Floor) -> Device*

Figura 2. Componentes Bucur et al. [17]

Otra aproximación para definir los componentes contextuales es la presentada por Kirsch-Pinheiro *et al.* [42]. Ellos proponen como componentes la ubicación del usuario (lugar), las herramientas que se usan (el sistema operativo, las aplicación, entre otras), el tiempo de conexión, la comunidad a la que pertenece el usuario, su rol dentro de la comunidad y el proceso que desempeña el usuario dentro de la comunidad.

3.1.1.1.2. Trabajos relacionados

El contexto de uso hace referencia a las características de la interacción del usuario con el sistema. Dey *et al.* [30] define el contexto como “*todos los elementos de información que pueden ser utilizados para caracterizar la situación de una entidad, siendo una entidad toda persona, lugar u objetos considerada pertinente para la interacción entre el usuario y la aplicación*”.

A continuación, se presenta una comparación de algunos trabajos relacionados con el contexto (ver Tabla 1). En los trabajos, se exponen diferentes perfiles de contexto y sus componentes, para más información sobre los trabajos relacionados ver Anexo C.

Los trabajos se encuentran enunciados en las columnas y en las filasse muestran los diferentes aspectos a comparar. A su vez se cuenta con las siguientes conversiones: *i)* el carácter “+” indica que el aspecto nombrado en la fila, se encuentra contemplado en el trabajo de la columna, *ii)* el carácter “-” simboliza que el aspecto no está considerado en el trabajo y por último *iii)* el carácter “?” alude a que el componente es nombrado, pero no se especifica cómo es tratado. Esta convención se utilizará en todas las tablas de comparación del presente trabajo de grado.

	[42]	[35]	[18]	[22]	[28]	[8]
Espacio temporal	+	+	+	+	?	?
Actividad	?	-	+	-	+	?
Cultural	+	-	-	-	?	-
Restricciones	-	+	-	-	?	-
Privacidad	-	+	+	-	+	-
Estructural	-	+	-	-	-	-
Términos relacionados	-	-	-	-	-	+
Comunidad	+	-	-	-	-	-

Tabla 1. Comparación de artículos de contexto

De la Tabla 1 se puede concluir, que los trabajos relacionados contemplan en su gran mayoría las características espacio temporales y a su vez se tienen en cuenta datos sobre la privacidad de la información a utilizar. Sin embargo, ninguno de los trabajos cumple con varios aspectos importantes a la hora de utilizar el contexto, como por ejemplo, el papel del usuario dentro de la comunidad. Este componente se encuentra en pocos trabajos y puede ayudar a definir los roles de las personas, las prohibiciones (leyes) a las que se ven sometidos, entre otros elementos importantes en la comunidad, por su parte las características estructurales del lugar donde se encuentra el usuario tampoco son tenidas en cuenta y podrían ser importantes en el momento de acceder a los *SI* por medio de conexiones inalámbricas que pudieran ser alteradas por las características físicas del lugar en el que se encuentra el usuario.

Habiendo descrito el contexto de uso, uno de los aspectos fundamentales a tener en cuenta para la adaptación al contenido, se procede a describir el perfil del usuario, donde se incluyen componentes como las características de los usuarios, sus preferencias y gustos, entre otros, con el fin de brindarle la información que mejor se ajuste a las características y necesidades de cada individuo en específico.

3.1.1.2. Perfil de usuario

Un usuario es quién busca información en los *SI*. El perfil de usuario es un modelo que busca representar las características de un usuario dentro del sistema, con el fin de adaptar información respecto a dichas características. Entre éstas se pueden mencionar los datos básicos, las preferencias, gustos y necesidades de cada usuario[21][57].

A su vez, la necesidad de modelar al usuario, surge al ver que cada uno puede tener objetivos e intereses diferentes, incluso un mismo usuario podría tener intereses diferentes dependiendo del contexto en el que se encuentre [71].

3.1.1.2.1. Componentes del perfil de usuario

Al igual que en el contexto, los elementos incluidos en el perfil de usuario varían dependiendo de los autores que lo propongan. En esta sección se procede a describir los componentes que han sido incluidos por diversos autores dentro del perfil de usuario.

Por ejemplo, Carmichael *et al.* [21] proponen un modelo de usuario definido a partir de su localización, su actividad actual y el Dispositivo de Acceso (*DA*) que está usando, teniendo en cuenta que el usuario puede haber definido preferencias para sus actividades. El sistema que proponen se denomina *Locator*, donde se utiliza dicho perfil. En *Locator*, el usuario puede definir por ejemplo, que mientras se encuentre en su oficina no desea recibir visitas físicas. El sistema sabría que el usuario se encuentra en su oficina, por la hora y actividad que está realizando. Si otro usuario solicita contacto con él, entonces el sistema le avisa al segundo usuario que éste no está disponible. También se establecen preferencias sobre a quién se notifica qué. Por ejemplo, si el segundo usuario es familiar del primero, el sistema alertará de la presencia de este y le dirá al familiar que puede seguir a la oficina.

A su vez, Tamine *et al.* [71] definen un perfil multidimensional, basado en dos dimensiones; una referente al histórico de las interacciones del usuario con el sistema y la otra a los intereses del usuario. En este trabajo se consideran el perfil conformado por la localización del usuario, la(s) aplicación(es) en uso, el tiempo de conexión y las preferencias de usuario.

Por su parte, *MAIPU* [57], (acrónimo de Modelo de Adaptación de Información basado en Perfil de Usuario), define un modelo con cuatro componentes: datos básicos, gustos, intereses y preferencias de usuario (ver Figura 3). Este modelo fue diseñado para ser utilizado para ideas de negocio. El componente de datos básicos contempla, el nombre del usuario, un identificador y la información que no cambia con el tiempo; los gustos por su parte, definen lo que un usuario en particular tiene como interés en un momento dado. Y se cuenta con un módulo de preferencias, pensadas especialmente para los negocios: las preferencias de los usuarios sobre productos y servicios (es decir para un negocio), actividades, resultados y actividades relacionadas.

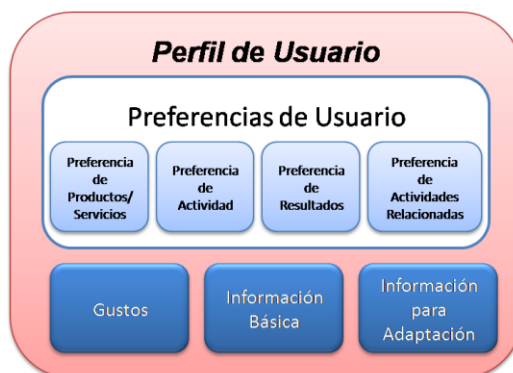


Figura 3. Perfil de usuario propuesto por MAIPU

Por otra parte se cuenta con un módulo grupal (*MAICO*) [23], que al igual que *MAIPU* fue desarrollado en el marco de un proyecto especial dentro de la Pontificia Universidad Javeriana. Este modelo de adaptación, se encarga de representar las comunidades y empresas, a través de las características más significativas y distintivas de grupos de individuos.



Figura 4. Arquitectura de *MAICO*

Como se puede observar en la Figura 4, *MAICO*¹² se divide en tres módulos: *i*) el módulo comunidad, que se encarga de modelar las preferencias y el comportamiento de una comunidad (es decir, un grupo de personas, unidas por temas, costumbres o patrones de comportamiento similares); *ii*) el componente empresarial, que busca modelar el comportamiento y las preferencias de una empresa, respecto a los procesos y actividades desarrolladas dentro de la misma y *iii*) cinco componentes compartidos, que son utilizados por los dos módulos anteriores, y que especifican la información general de la comunidad, como el nombre de la misma, sus miembros y su ubicación; además, describen las políticas internas, las relaciones con otras comunidades, la posición jerárquica en la que se encuentra dicha comunidad, si es una comunidad institucional o no, y las preferencias grupales.

¹² Para ver el documento completo de *MAICO*, remítase al anexo C

3.1.1.2.2. Trabajos relacionados

Habiendo definido el perfil de usuario, se procede a presentar una tabla, con los componentes utilizados en algunos trabajos que proponen o usan un perfil de usuario, como modelo de adaptación al contenido. Para más información sobre estos trabajos, diríjase al Anexo C.

En la Tabla 2, se muestra una comparación entre los datos tenidos en cuenta en cada uno de los perfiles de usuario. Los trabajos relacionados se encuentran enunciados en las columnas, y en las filas se muestran los diferentes aspectos a comparar. A su vez se cuenta con las mismas conversiones utilizadas para la tabla de perfil contextual.

	[21]	[57]	[32]	[23]	[78]	[48]	[11]	[52]
Localización	+	?	?	-	-	-	?	-
Preferencia actividad	+	+	-	-	-	-	+	?
Dispositivo de acceso	+	?	-	-	?	-	?	?
Preferencias	?	+	-	-	+	-	+	-
Productos	-	+	-	+	-	-	?	-
Preferencias de Resultados	-	+	-	-	-	-	?	-
Información relacionada	-	+	-	+	-	-	+	-
Gustos	?	+	-	-	+	-	+	?
Información básica	?	+	?	+	?	?	+	+
Estado	-	-	+	-	-	-	?	-
Aplicación	-	-	-	-	-	+	?	+
Tiempo	-	?	+	-	-	+	?	+
Seguridad	-	-	-	-	-	+	?	+

Tabla 2. Comparación trabajos perfil de usuario

Como se puede ver en la anterior tabla, los perfiles de usuario propuestos en los diversos artículos, consideran aspectos que incluyen desde la información básica del usuario (nombre, identificación, teléfono, entre otros), gustos, sus preferencias de actividad hasta preferencias de resultados (indican cómo desea el usuario recibir la información). Sin embargo, la mayoría de los trabajos no incluyen características de seguridad, que podría ser definidas como preferencias de quiénes pueden acceder a cierta información o realizar ciertas funciones o solicitar servicios dentro del sistema.

3.1.1.3. Formas de representación de los perfiles

Así como los componentes que definen el perfil contextual difieren dependiendo de los autores; las representaciones de los mismos también lo hacen. Por ejemplo, pueden ser representados por medio de: *XML*, modelos orientados a objetos, ontologías, entre otros.

Tal es el caso de Ruiz *et al.* [65], quienes proponen un modelo de representación usando ontologías. Estas pueden ser fuentes de información para los *SI*, es decir, que las ontologías son enviadas a los *SI* y éstos las utilizan según sus necesidades. Los sistemas que utilizan ontologías como fuentes de información son conocidos como sistemas *sensibles a las ontologías*.

Por su parte Kirsch-Pinheiro *et al.* [42] proponen un sistema para la representación del contexto mediante un diseño orientado a objetos. Dicho perfil está compuesto por *i)* un módulo espacial, que hace referencia a dónde se encuentra el usuario, es decir una localización física; *ii)* un módulo temporal, que hace referencia a un calendario, incluyendo el día, el mes y el año; *iii)* un módulo de comunidad, que indica cómo es el grupo donde se encuentra el usuario y su rol dentro de la comunidad; y *iv)* el proceso, que se refiere a las actividades realizadas dentro de la comunidad. En la Figura 5 se puede ver éste modelo orientado a objetos, los elementos que conforman el contexto, las relaciones con sus elementos y las cardinalidades.

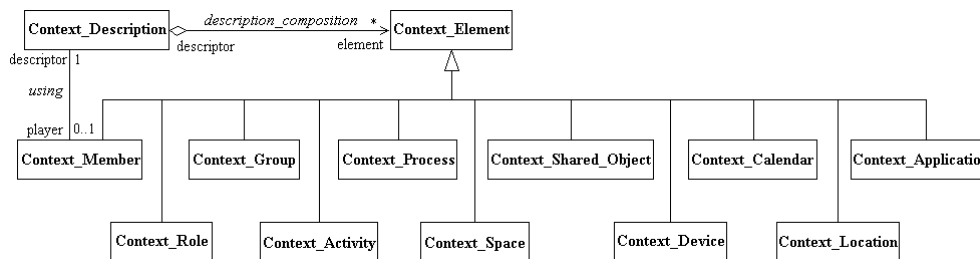


Figura 5. Representación de contexto, tomado de [42]

A partir de lo anterior, se puede afirmar que las ontologías poseen una estructura estándar, es decir, que para describir un concepto siempre se parte de lo general a lo particular, por lo cual una ontología es entendible para personas que no tengan gran conocimiento sobre este tipo de representación. Las representaciones orientadas a objetos por su parte, definen los componentes y las relaciones (composición, agregación, herencia) que existen entre ellos, así como la multiplicidad de las mismas. Para entender los diagramas orientados a objetos (*e.g.* diagrama de clases, de dominio, entre otros) es necesario que las personas tengan un grado de conocimiento de *UML*, más específicamente, sobre las notaciones de los diagramas y los conceptos del paradigma de programación orientada a objetos. Como se puede ver, cada una de las representaciones tiene ciertas ventajas, por ejemplo, *XML* es un lenguaje de fácil interpretación para las diferentes fuentes de información, los modelos orientados a objetos muestran las relaciones que existen entre los componentes, mientras que las ontologías poseen una estructura estándar que comienza en lo más general para ir derivando en lo específico. Para mayor información de formas de representación de perfiles ver Anexo C.

3.1.1.4. Adquisición de información para los perfiles

Kassab *et al.* [39] definen que la información puede ser adquirida por medio de: *i*) interfaces, en las cuales los usuarios ingresan los datos, *ii*) perfiles predefinidos y *iii*) datos consignados en registros de uso, construidos a partir de los datos que ingresan los usuarios en diferentes sesiones, es decir los datos históricos.

Para la adquisición de datos contextuales, se pueden utilizar sensores incluidos en el dispositivo de acceso (*DA*), como el *GPS*, que indica la ubicación del usuario. A su vez, se utilizan los métodos nombrados por Kassab *et al.* [39], cuando no se cuenta o no se pueden utilizar los sensores. Tal es el caso de las prioridades, para las cuales no se puede obtener la información por medio de sensores, entonces se podría utilizar una interfaz gráfica.

Para la adquisición de información mediante datos históricos, existen algoritmos de minería de datos [60][76][25]. Estos algoritmos son utilizados para encontrar patrones de comportamiento en los datos e información potencialmente útil (no conocida), a partir de lo consignados en una en bases de datos, en la Web, o en otros grandes repositorios de información; para esto se pueden utilizar modelos matemáticos y/o estadísticos, lógica difusa, árboles de decisión, algoritmos genéticos, entre otros. A partir de la minería, se podría obtener la información que sea necesaria para el contexto actual, por ejemplo, se podría inferir cómo va a estar el clima un día determinado dependiendo de cómo estuvo los días anteriores, o también se podrían inferir las consultas que realiza normalmente un usuario, al encontrarse realizando una actividad determinada o al estar en un lugar determinado.

Para mayor información sobre la adquisición de la información, remítase al Anexo C.

3.1.1.5. Priorización de componentes

Algunos autores se han preguntado cómo escoger los componentes a ser tenidos en cuenta dentro de la adaptación, debido a la importancia que éstos tienen, tanto para el usuario, como para los fines de la aplicación (*e.g.*, una aplicación sensible a la localización dará prioridad al componente que maneje la ubicación, sobre uno que maneje la parte estructural). A partir de esta importancia, se han definido diversos métodos de priorización algunos de ellos son enunciados a continuación.

Panayiotou *et al.* [58] proponen un método de priorización, para establecer las preferencias de los usuarios sobre los componentes. Para esto, se le asigna a cada componente un número entre

-1 y 100. Cuando se asigna el valor de -1, se indica que definitivamente no se tendrá en cuenta el componente. Un valor más alto indica que el componente que la tiene será prioritario sobre otro que tenga un menor valor. En la Figura 6, se muestra parte de un perfil que incluye un ejemplo de prioridades establecidas para tipos de comida según la hora del día. Dichas prioridades influyen directamente en el conjunto de resultados que será presentado al usuario. Si hay alguna que tenga un peso de -1, ésta será una preferencia no tomada en cuenta.

```
</timeZone>
<timeZone time="3-5"></timeZone>
<timeZone time="6-9"></timeZone>
<timeZone time="9-12">
  <restaurantType name="Fast Food" weight="90"/>
  <restaurantType name="Greek" weight="70"/>
</timeZone>
```

Figura 6. Perfil de preferencias con prioridades establecidas.

El criterio para la asignación de los valores varía dependiendo de la aplicación y de la información ingresada por los usuarios. La priorización se realiza con el fin de definir los componentes contextuales que deberían ser tenidos en cuenta. Dey *et al.* [26] proponen que el contexto a ser utilizado y las prioridades del mismo, depende del tipo de aplicación que lo utilizará. En la Figura 7, se muestra la propuesta de los elementos a ser tenido en cuenta, según el tipo de aplicación a usar. Se utilizan las siguientes convenciones: “A” representa el componente actividad, “I” el identificador de la persona, “L” es la localización y “T” el tiempo.

System Name	System Description	Context Type			
		A	I	L	T
Classroom 2000 [1]	Capture of a classroom lecture			X	X
Cyberguide [1]	Tour guide		X	X	
Teleport [2]	Teleporting	X	X	X	
Stick-e Documents [3,4,5]	Tour guide		X	X	X
	Paging and reminders	X	X		
Reactive Room [6]	Intelligent control of audiovisuals	X	X	X	
GUIDE [7]	Tour guide			X	
CyberDesk [8,9,10]	Automatic integration of user services	X			
Conference Assistant [11]	Conference capture and tour guide	X	X	X	X
Responsive Office [12]	Office environment control			X	X
NETMAN [13,16]	Network maintenance			X	
Fieldwork [17,18,22]	Fieldwork data collection			X	X
Augment-able Reality [19]	Virtual post-it notes			X	
Context Toolkit [24]	In/Out Board		X	X	X
	Capture of serendipitous meetings		X	X	X
Active Badge [28]	Call forwarding		X	X	

Figura 7. Priorización de componentes tomado de [26]

Teniendo claro qué son los perfiles de usuario y de contexto, cuáles son los componentes que se incluyen en estos, y considerando que la información proveniente de dichos perfiles es la que se utiliza para adaptar el contenido que se entrega a un usuario, en el momento en que éste realiza una consulta, surge la necesidad de explicar el proceso que se lleva a cabo para identificar las fuentes de información que puedan responder la consulta, y la forma en que los resultados son

entregados nuevamente al usuario. La siguiente sección explicará el proceso de enrutamiento de consultas.

3.2. Enrutamiento de consultas

La información que se puede encontrar en las diferentes Fuentes de información (*FI*) crece de manera continua y el tratamiento de dicha información se vuelve cada vez más complejo [26]. Es por eso que existen diferentes formas de llevar a cabo los procesos de selección de *FI* que puedan contestar una consulta, como son: la agrupación de pares según la información que manejan [11][43], los histogramas [62], sistemas de confianza y reputación [3], correspondencia de datos [44], entre otros. De esta forma, se busca saber qué información maneja cada uno de los nodos o *FI*, con el fin de enviarle la consulta y esperar/consolidar los resultados. Este proceso es llamado *Enrutamiento de Consultas (EC)*. El enrutamiento de consultas es el mecanismo por el cual se ubican la(s) *FI* que puede(n) satisfacer¹³ una determinada consulta[59]. Más específicamente se definen el *EC* como la propagación de las consultas del usuario hacia las fuentes de informaciones más apropiadas [59][67].

A continuación se describirán las actividades que pertenecen al proceso de enrutamiento de consultas.

3.2.1. Actividades que involucra el proceso

Como se mencionó anteriormente, el *EC* es un proceso que involucra ciertas actividades. A continuación se mencionarán las actividades que diferentes autores consideran necesarias para realizar el enrutamiento.

En *GoogleGuide*, un tutorial interactivo para el manejo del motor de búsqueda *Google*[®] (http://www.googleguide.com/google_works.html), se contempla los siguientes pasos: *i*) El usuario ingresa la consulta que desea y la envía usando la página web de *Google*[®]; *ii*) la consulta llega al servidor web de *Google*[®], el cual redirige al servidor de índices, que indica cuáles páginas contienen los términos dados en la consulta. *iii*) Las referencias encontradas por el servidor de índices, son redirigidas al servidor de documentos, que construye un conjunto de resultados a partir de extractos que tiene de cada página indexada, *iv*) el conjunto de resultados es enviado al usuario que hizo la consulta.

Tempich *et al.* [73] muestran que el *EC* se compone de las siguientes actividades: *i*) la selección de los pares que contestarán la consulta, en donde se seleccionan los pares que tienen

¹³ Satisfacer una consulta, hace referencia a contestarla total o parcialmente.

mayor probabilidad de contestarla; si no se puede seleccionar ninguno, la consulta es enviada a todos los pares. A los pares que fueron seleccionados se les envía un mensaje que contienen un identificador (*id*) de consulta y un *id* de los pares que se han visitado, con el fin de evitar la presencia de ciclos. Luego de esto, *ii*) los pares verifican si tienen información con la cual se puede contestar la consulta, si es así, ésta es retornada. Para finalizar se realiza *iii*) una evaluación, en la cual se refina la clasificación de los pares, dependiendo de si dieron respuestas a la consulta y cuan relevante¹⁴ fue esa respuesta.

Así mismo, otros autores utilizan las tablas hash para el enrutamiento. En estas tablas se guarda el identificador del nodo y las palabras claves de los documentos que maneja. Cuando una consulta es introducida por un usuario, se comparan las palabras de la consulta con las consignadas en tabla; luego, la consulta es dirigida a cada par en el cual se encontraron equivalencias semánticas [40].

A continuación se presentan algunos trabajos relacionados con actividades del proceso de enrutamiento de consultas.

3.2.2. Análisis de consultas

Grine *et al.* [31] especifican el análisis de consultas como el proceso de convertir las consultas en datos manipulables a los cuales se les puedan realizar operaciones (*e.g.*, agregar caracteres, comparaciones, entre otros). Dependiendo del tipo algoritmo que se utilice para la selección de las fuentes de información se procede a enviar la consulta en forma total o a dividirla para que diferentes fuentes de información puedan responder adecuadamente. En algunos casos, se ejecuta primero la consulta de forma completa y si esta no trae los suficientes resultados se realiza la búsqueda dividiendo algunos de los términos como si fueran consultas más pequeñas y enviándolos así a diferentes fuentes de información [35][72].

Además de cómo se trata la consulta, también se debe tener en cuenta el proceso que se debe llevar a cabo para que una consulta sea respondida. Dicho proceso debe tratar de administrar los recursos del sistema donde se ejecute en tratamiento de la consulta para poder entregarle al usuario resultados correctos, y en el menor tiempo posible.

¹⁴ La relevancia es indicada por un algoritmo, que define las respuestas que serán entregadas al usuario, así como el orden de las mismas.

El procesamiento de consultas consta de cinco pasos (ver Figura 8). En primera instancia el analizador de consultas recibe una consulta de entrada, la cual es traducida en una consulta equivalente (es decir que se reescribe), de manera que ésta pueda ser ejecutada de una manera más eficiente. El término eficiencia se refiere a que las consultas sean contestadas en un tiempo menor, utilicen menos recursos de lo usual o que entreguen resultados más acordes a las necesidades del usuario. Al tener esta consulta optimizada, se procede a crear un plan de ejecución, que busca minimizar el uso del *CPU*, las entradas y salidas, y los costos de comunicación. Luego, se procede a ejecutar la consulta y obtener los resultados de la misma.

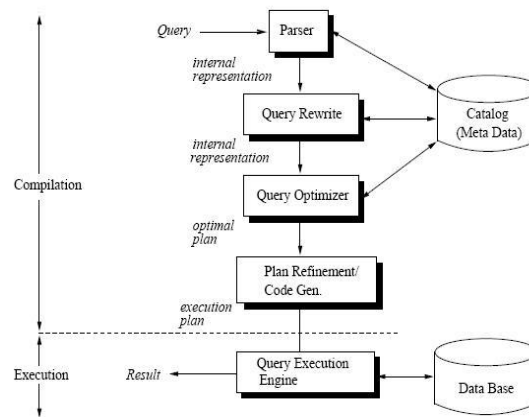


Figura 8. Análisis de consulta [29]

Por otra parte, se cuenta con consultas que se basan en un conjunto de *Palabras claves* o frases. Cuando la consulta es enviada a las fuentes de información para que éstas la contesten, se divide en palabras claves. Por ejemplo, una consulta que sea “vacaciones isla Java”, se traduce como tres palabras claves: “vacaciones”, “isla”, “Java”. Los resultados entregados, en primera instancia serán documentos que contengan las tres palabras claves; si no se encuentran resultados, se procede a buscar los documentos que posean algunas de las palabras claves [47], es decir: *i*) “vacaciones”, “isla”; *ii*) “vacaciones”, “Java”; *iii*) “isla”, “Java”; *iv*) “vacaciones”; *v*) “isla”; ó *vi*) “Java”. Así mismo, los motores de búsqueda como *Google*¹⁵, utilizan operadores con los cuales el usuario puede indicar si desea que todas las palabras claves sean incluidas en los documentos (palabra clave 1 *AND* palabra clave 2), si desean documentos que contengan cualquiera de las palabras claves (palabra clave 1 *OR* palabra clave 2); por defecto *Google*[®] utiliza el operador *AND* para sus búsquedas.

¹⁵ Explicación disponible en <http://www.googleguide.com>

En este punto cabe aclarar que, si un mecanismo de enrutamiento permite conocer qué *SI* son capaces de responder una consulta, también los sabrá si una la consulta se le agregan criterios¹⁶. Para enrutar una consulta con más criterios, no se deben agregar funcionalidades al enrutamiento, dado que el proceso de enrutamiento no debe variar de una consulta a otra [34].

3.2.3. Reescritura de consultas

Una consulta es una solicitud de información que va destinada a unas fuentes de información y está definida como un conjunto de tuplas de atributo y valor, $Q = \{ \langle a_1, v_1 \rangle \dots \langle a_n, v_n \rangle \}$, donde se definen las palabras claves a buscar [36][80].

En primera instancia, cabe aclarar que la reescritura de consultas consiste en transformar una consulta original (*Q*), en una consulta equivalente pero con mejor desempeño¹⁷. El proceso de reescritura y de determinar si la consulta es consistente es un problema del tipo *NP* [49][27][19]. La reescritura incluye una modificación sintáctica y semántica, es decir, se pueden cambiar conceptos por otros similares y que sean menos ambiguos, así como realizar cambios debido al contexto en el que son utilizadas o incluir operadores lógicos como “*OR*”, “*AND*”, entre otros [27].

3.2.3.1. Consultas *SQL*

Una consulta *SQL*, está conformada por tres cláusulas básicas: *i)* *SELECT*, es la parte de la consulta donde se indican los atributos que se desean consultar, *ii)* *FROM*, donde se especifican las tablas de donde se buscarán los resultados y *iii)* la cláusula *WHERE*, donde se detallan ciertas condiciones que se deben cumplir para las relaciones de la cláusula *FROM* [64].

3.2.3.1.1. Operadores Clausula *WHERE*

A continuación se presentan los operadores utilizados en la cláusula *WHERE* para especificar las condiciones que se tendrán en cuenta en el momento de buscar los atributos (ver Tabla 3).

¹⁶ Se pueden agregar criterios a las consultas, en busca de adaptar la información de los resultados. Estos criterios dependerían de las necesidades, características del usuario y el contexto de uso.

¹⁷ Para el caso del sistema de enriquecimiento, se busca realizar reescritura con el fin de personalizar los resultados que se entregarán al usuario a partir de una consulta y que dichos resultados satisfagan las necesidades de información del usuario y las condiciones de su interacción con el sistema.

Operador	Operación	Ejemplo
=	Igualdad	SELECT * FROM emp WHERE eid = 100;
!=, <>, ^=	Desigualdad	SELECT * FROM emp WHERE eid != 100;
<	Menor que	SELECT * FROM emp WHERE eid < 200;
>	Mayor que	SELECT * FROM emp WHERE eid > 200;
<=	Menor o igual que	SELECT * FROM emp WHERE eid <= 200;
>=	Mayor o igual que	SELECT * FROM emp WHERE eid >= 200;
IN	Igual a cualquiera de los miembros entre paréntesis	SELECT * FROM emp WHERE eid IN (100, 300);
NOT IN	Distinto a cualquiera de los miembros entre paréntesis	SELECT * FROM emp WHERE eid NOT IN(200);
BETWEEN	Contenido en el rango (números, texto o datos)	SELECT * FROM emp WHERE eid BETWEEN 99 AND 200;
NOT BETWEEN	Fuera del rango	SELECT * FROM emp WHERE eid NOT BETWEEN 100 AND 199;
LIKE '_abc%'	Contiene la cadena 'abc' a partir del segundo carácter y luego cualquier cadena de caracteres	SELECT * FROM emp WHERE ename LIKE 'AABC%';
AND	Muestra los registros si la primera y la segunda condición es verdadera	SELECT * FROM emp WHERE eid >200 AND eid <300;
OR	Muestra los registros si la primera o la segunda condición es verdadera.	SELECT * FROM emp WHERE eid = 200 OR eid = 300;
NOT	Muestra los registros en donde no se cumple la condición expuesta	SELECT * FROM emp WHERE NOT eid = 300;
IS NULL	Se utiliza para comprobar los valores nulos	SELECT * FROM emp WHERE eid IS NULL;
IS NOT NULL	Se utiliza para comprobar los valores no nulos	SELECT * FROM emp WHERE eid IS NOT NULL;

Tabla 3. Operadores de comparación, tomado de [59]

3.2.3.2. Trabajos relacionados con la reescritura de consultas

A partir de los trabajos relacionados con la reescritura de consultas, se presenta una tabla comparativa de los mismos. Para más información sobre estos trabajos remítase al Anexo C.

	[27]	[80]	[7]	[41]	[49]
Reglas lógicas	+	+	+	?	+
Consultas históricas	-	+	-	+	-
Reglas de equivalencia	-	?	+	?	-
Semántica	+	+	+	+	+
Sintaxis	+	+	+	+	-
Minería de datos	-	+	-	?	-

Tabla 4. Comparación de trabajos de reescritura de consultas

Como se observa en la Tabla 4, la mayoría de los trabajos utilizan reglas lógicas para realizar la reescritura de consultas, y muchas veces se tienen en cuenta similitudes semánticas y/o sintácticas para modificar los términos que poseen las consultas. En cambio, se puede observar que casi ninguno de los anteriores trabajos considera algoritmos de minería de datos para realizar tal reescritura.

3.2.4. Selección de fuentes de información

Para el enrutamiento de consultas (*EC*) se cuenta con diversos métodos para seleccionar las fuentes de información (*FI*) que puedan satisfacer total o parcialmente las consultas.

Las *FI* pueden ser divididas dependiendo de los servicios que prestan y la información que manejan; a esto se le llama agrupamiento de pares [15].

Jin *et al.* [38] proponen un algoritmo que se encarga de realizar un agrupamiento de pares por categorías (ver Figura 9), dependiendo del tema que estos manejan. Para realizar dicho agrupamiento se realiza primero una fase de entrenamiento, donde se incluyen unos documentos con una categoría conocida. A partir de esto, se realiza la clasificación inicial, luego los documentos son representados en un vector, y son clasificados en diferentes categorías. Cuando el sistema se inicia, cada par envía la información de las categorías que maneja, y cuando una consulta es dirigida a este sistema de pares, se selecciona un par que sea semánticamente similar a los términos de la consulta. Para determinar los pares similares semánticamente, se utilizan las palabras claves de la consulta, las cuales son comparadas con los temas manejados por los pares.

Input: Query request Q , Semantic topology SP_i of Peer P_i , Document set D_i with category labels.
Output: Result R similar to Q , Peers collection PF to be forwarded.

```

1:  $PF := \Phi$ 
2: if  $Q$  is keyword exact-matching request then
3:   perform local keyword search and return  $R$ 
4:   random select known peers  $PF$  to forward  $Q$ 
5: end if
6: else if  $Q$  contains semantic search fields then
7:   predict  $Q$ 's category  $Q_c$ 
8:   perform local search and return  $R$ 
9:   for each peer  $p \in SP_i$  do
10:    calculate similarity  $\text{sim}(Q_c, C_p)$ 
11:    if  $\text{sim}(Q_c, C_p) > \text{Threshold}$  then
12:      put  $p$  into set  $PF$ 
13:    end if
14:   end for
15:   forward  $Q$  to peers collection  $PF$ 
16: end if

```

Figura 9. Enrutamiento de consultas, tomado de [38]

Para realizar el enrutamiento de consultas, cada par puede contener un histograma que es llamado *índice local* y que posee la cantidad de información manejada por cada temática; los histogramas, son entonces usados para seleccionar los pares que responderán la consulta y para balancear las cargas de los mismos. En este caso, para encontrar las fuentes de información que puedan responder la consulta se realiza un algoritmo en el cual los pares que reciben la consulta la propagan a los pares vecinos, que dentro en su histograma muestran si están en capacidad de proveer resultados. Este algoritmo finaliza cuando se ha recorrido un número máximo de pares definido previamente o cuando se han encontrado un número de resultados satisfactorios (también es un número predefinido). Lo primero que se hace en cada par, es verificar si el índice local contiene datos que pueden responder total o parcialmente la consulta, si es así, estos resultados son retornados; luego se procede a verificar si el número máximo de pares recorridos es igual al número que se estableció como máximo o si ya se han encontrado el número de resultados esperados, si alguna de estas dos condiciones se cumple, la búsqueda termina. Si no se cumple alguna de estas condiciones, se propaga la consulta a los pares que puedan responderla según su contenido local [62].

La selección de fuentes de información propuesta por Tempich *et al.* [72] consta de un algoritmo de selección, un algoritmo de clasificación de pares y una evaluación de los resultados brindados por los pares. En el algoritmo de selección se escogen los pares que probablemente puedan contestar la consulta del usuario, para esto se tienen un repositorio local, donde se evalúan los criterios de la búsqueda con los datos que maneja cada par; así mismo, se utiliza un algoritmo de clasificación, para determinar la forma en que se agrupan los pares dependiendo de la información que manejen, y a partir de esto se seleccionan las fuentes de información para una consulta determinada. Si la cantidad de pares seleccionados es menor a un máximo establecido (P_{\max}), se procede a ejecutar un algoritmo de relajación; en este algoritmo se procede a realizar la búsqueda excluyendo algunos de sus términos, con el fin de “relajar” la consulta y poder obtener más pares que puedan contestarla. Por ejemplo, si se busca “java isla

indonesia” y no se encuentran la cantidad de pares esperados, se utiliza el algoritmo de relajación que buscará el conjunto de términos: “java isla” o “java indonesia” o “isla indonesia” o cada término por separado (“java” o “isla” o “indonesia”), para poder encontrar una mayor cantidad de pares (ver Figura 10).

<hr/> Algorithm 1 Peer Selection: $\text{peerSelection}(O, Q, A)$ Require: LocalNodeRepository O , Queue of Queries Q , Queue of peers A 1: $Q_{relaxed} := \emptyset$ 2: Queue $\text{selectedPeers} := \emptyset$ 3: for all $Q \in Q$ do 4: $S_Q := \text{performQuery}(O, Q)$ 5: for all $S \in S_Q$ do 6: $MO := \text{retrieveAllMetadata}(O, S)$ 7: for all $MO \in MO$ do 8: $\text{selectedPeers.push}((MO.\text{peer}, MO.RC))$ 9: end for 10: end for 11: $Q_{relaxed} := Q_{relaxed} + \text{relaxQuery}(O, Q)$ 12: end for 13: $A := A.\text{append}(\text{rankPeers}(\text{selectedPeers}))$ 14: if $ A < p_{max}$ then 15: $A := \text{peerSelection}(O, Q_{relaxed}, A)$ 16: end if 17: return A <hr/>	<hr/> Algorithm 2 Peer Rating: $\text{rankPeers}(\text{selectedPeers})$ Require: Queue of pairs selectedPeers 1: Set $\mathcal{P} := \{P \exists RC : (P, RC) \in \text{selectedPeers}\} = \{P_1 \dots P_n\}$ 2: for all $P \in \mathcal{P}$ do 3: $\mathcal{RC} := \{RC (P, RC) \in \text{selectedPeers}\}$ 4: for all $RC \in \mathcal{RC}$ do 5: $\text{strength}(P) := tc \cdot P.OC + (1 - tc) \cdot \frac{1}{ \mathcal{RC} } \sum_{RC \in \mathcal{RC}} RC$ 6: end for 7: end for 8: Queue $\text{rankedPeers} := (P_1, \dots, P_{ P })$, where $\text{strength}(P_1) \geq \dots \geq P_{ P }$ 9: RETURN rankedPeers <hr/> Algorithm 3 Query Relaxation: $\text{relaxQuery}(O, Q)$ Require: O, Q $\text{state} := \text{determineState}(Q)$ $Q := \text{newQuery}(O, Q, \text{state})$ return Q <hr/>
--	---

Figura 10. Selección y clasificación de pares [72]

3.2.5. Creación de cadenas en un buscador

Cuando una consulta es ingresada a un motor de búsqueda, éste realiza el proceso de enrutamiento a partir de palabras relacionadas con las ingresadas en las consultas, pero se hace solamente por concordancia. En el caso de *Google*^{®18}, se define una consulta como una petición de información a un motor de búsqueda. En estas consultas, se deben seleccionar palabras que tengan relación con los resultados que se desean obtener.

Para la selección de fuentes de información, *Google*[®] hace un proceso de tres fases. En la primera un robot informático llamado *Googlebot*, busca páginas en la red y luego se las entrega al indexador, que clasifica las palabras existentes dentro de la página y las almacena en una base de datos. Luego, el procesador de consultas se encarga de tomar las consultas que han sido generadas por los usuarios, las compara con el índice y trae como resultado las páginas que considera relevantes. La relevancia o no de las páginas está dada por el algoritmo PageRank^{®19}, que establece para cada página indexada, un número asociado con la cantidad de enlaces que apuntan a ésta.

¹⁸ Explicación disponible en <http://www.googleguide.com>

¹⁹ Descripción disponible en <http://www.ianrogers.net/google-page-rank/>

A nivel de *SQL*, se cuenta con trabajos como el de Arocena *et al.* [6], quienes proveen una descripción del lenguaje *Websql*, que plantea una unión entre las bases de datos relacionales que usan el lenguaje de consultas *SQL*, y los motores de búsqueda en la web. Con este lenguaje es posible armar consultas tipo *SQL* que realizan búsquedas sobre las páginas web. Por ejemplo, *Such that*, que significa “tal que”, que se aplica para páginas que cumplan cierta condición (es análogo a *Where*, término propio de *SQL*).

3.2.6.Recepción de resultados

Luego de realizar consultas a los sistemas de información (*SI*), se recibe un conjunto de resultados que se deben mostrar al usuario. Al recibirlos, se le pueden mostrar tal y como llegan, o seleccionar los que cumplen algún criterio, es decir filtrar los resultados [48].

Los resultados que se reciben de los *SI* con técnicas de *IR* (*information retrieval*), se pueden filtrar para entregar a los usuarios lo que estos consideran pertinente. Esta consideración de pertinencia es dada por el usuario.

Shapira *et al.*[61] describen que un sistema de filtrado básico, está compuesto por *i*) un analizador de información, el cual es el encargado de adquirir información de los proveedores; esta información es analizada y enviada a un *ii*) componente que se encarga del proceso de filtrado. A su vez, se cuenta con *iii*) un perfil de usuario, que contiene la información que permite identificar las necesidades del usuario; esta información es enviada al proceso de filtrado. Luego, en el proceso de filtrado se compara la información del perfil de usuario y del analizador de información, con el fin de identificar los resultados que podrían ser relevantes para los usuarios. Para finalizar, el usuario realiza una evaluación sobre los resultados retornados, que será enviada *iv*) al proceso de aprendizaje, el cual es el encargado de actualizar el perfil de usuario (ver Figura 11).

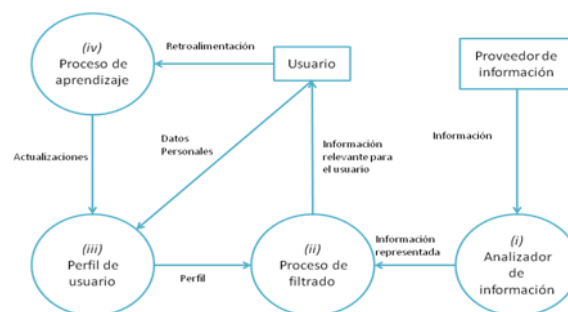


Figura 11. Sistema de filtrado, adaptado de[61]

Basado en esta información, el sistema de filtro que proponen Li *et al.* [46], utiliza un perfil de usuario, esto con el fin de entregar al usuario información acorde a sus gustos. Luego de

entregar los resultados al usuario, se realiza una retroalimentación, que busca determinar si los documentos entregados fueron pertinentes o no; esta información será considerada en futuras búsquedas, para evitar entregar documentos que no pertinentes para el usuario.

El trabajo de Arezki *et al.* [5] propone representar en forma de grafo las características comunes existentes en un conjunto de documentos. Esta estructura de datos relaciona palabras clave con la cantidad de ocurrencias que tienen en un documento. De acuerdo con la cantidad de veces que aparecen las palabras claves de la consultas del usuario, en los documentos, se establece la prioridad entre ellos, siendo más alta la del documento que más veces contiene las palabras claves. Sin embargo, no se contemplan otras características del usuario, se identifica a éste con un número único, pero no se le asocian más características, es decir, no se tienen en cuenta ni datos generales, ni preferencias, ni gustos. Luego de definir cómo se lleva a cabo el proceso para contestar las consultas suministradas por un usuario, se procede a explicar el enriquecimiento de consultas, que tiene como fin último adaptar las consultas, para entregar al usuario información acorde a sus gustos, preferencias y necesidades.

3.3. Enriquecimiento de consultas

Las consultas realizadas por los usuarios, generalmente son cortas y poco específicas. A pesar de eso, los usuarios esperan que su búsqueda arroje el resultado que ellos quieren, y que estos se encuentren en los primeros diez resultados mostrados [24].

Cuando un usuario solicita información a un sistema de información, éste realiza una consulta (Q), en la cual escribe los datos claves que considera podrán describir su solicitud, por ejemplo si un usuario desea obtener información sobre algunos restaurantes en Bogotá, su consulta podría ser “Restaurantes Bogotá”. Pero sin embargo, los usuarios realizan consultas cortas y poco específicas, lo que causa en algunos casos que los resultados sean inoportunos con respecto a las necesidades del usuario. En el caso del ejemplo anterior, se podrían presentar restaurantes que por su costo no se ajusten al presupuesto del usuario o que sean de algún tipo de comida que no le agrada, generando problemas como la sobrecarga cognitiva al tener resultados de comidas que no le gustan al usuario.

Así mismo, es importante considerar aspectos que el usuario no indica en su consulta, como sus preferencias y el contexto que lo rodea. Por ejemplo, el usuario que realizó la consulta anterior Q (“Restaurantes Bogotá”) cuenta con un perfil, donde se establece que su comida favorita es la italiana y un perfil contextual que indica que actualmente se encuentra ubicado en la

universidad. Según estos datos, el usuario podría buscar los restaurantes en Bogotá que ofrecen comida italiana y que estén dentro de la universidad. A la consulta inicial se le incluyen estos datos, con los cuales se podría convertirla en “Restaurantes universidad Javeriana Bogotá comida italiana”, con lo cual se podrían satisfacer los deseos del usuario respecto a sus gusto y su ubicación actual, aspectos que normalmente son descartados por los sistemas convencionales de búsqueda de información. Esta nueva consulta recibe el nombre de consulta enriquecida (Q'). Es decir, una consulta enriquecida es generada a partir de una consulta inicial (Q), a la que se le incluyen ciertos parámetros de acuerdo con las necesidades de información del usuario. Mediante este enriquecimiento, lo que se busca es poder brindar resultados con los cuales el usuario sienta que el sistema está diseñado según sus necesidades [83].

El enriquecimiento de consultas, entonces se puede definir como el proceso de transformar una consulta Q en una consulta Q' .

3.3.1. Actividades que pertenecen al proceso

Bai *et al.* [8] buscan realizar un enriquecimiento de consultas basado en las características contextuales. Para dicho enriquecimiento se tienen los siguientes pasos: *i*) se define un modelo, el cual se refiere a un conjunto de términos relacionados con un tema; estas relaciones son definidas a partir un área de conocimiento y pueden ser establecidas manualmente por un usuario, o por medio de procesos matemáticos. Por ejemplo, “Programación, Java” estaría relacionado, en este caso con el área de conocimiento “Computación” y la relación se definiría: “{Programación, Java}” \rightarrow “Computación”. Estas relaciones se definen a modo de reglas, donde se considera que si el o los términos de la izquierda se encuentran en la consulta entonces la parte derecha será considerada para el enriquecimiento; *ii*) se define un dominio, es decir, un conjunto de términos pertenecientes a un tema en general, por ejemplo “polución”, “lluvia” y “temperatura” podrían hacer parte del dominio “ambiente”. Para finalizar, *iii*) se agrega el tema del dominio y los términos relacionados a la consulta inicial. Sin embargo, este trabajo no considera características del usuario que pueden ser importantes a la hora de decidir con qué términos o temas, relacionar las consultas de los usuarios. Además, se podría ampliar la definición de contexto para tener en cuenta otras características, puede ayudar a que el tema con el que se relacionan sea el indicado, dependiendo del momento en el que se realiza la consulta (el tema con el que el usuario desea relacionar su consulta puede ser distinto en dos momentos diferentes del día).

Storey *et al.* [69] definen, en primera instancia los pasos que se tienen que seguir al realizar búsquedas en la Web (ver Figura 12), y a partir de éste se define el proceso de enriquecimiento

de consultas. Este proceso está conformado por los siguientes pasos: *i*) identificar cuál es la información que el usuario necesita, *ii*) modificar la consulta inicial del usuario, es decir, agregar los datos necesarios dependiendo de las necesidades del usuario, *iii*) ejecutar la consulta, *iv*) evaluar si se obtuvieron resultados; de no ser así se modifica nuevamente la consulta.

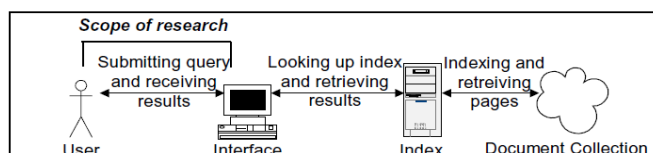


Figura 12. Proceso de búsquedas Web [69]

A continuación se presentan algunos trabajos relacionados con el enriquecimiento de consultas.

3.3.2. Trabajos relacionados

Luego de revisar algunos trabajos relacionados con el enriquecimiento de consultas, se presenta una tabla comparativa, en las columnas se presentan los componentes que se tienen en cuenta para el enriquecimiento y en las columnas los trabajos relacionados (ver Tabla 5). Para más información sobre estos trabajos remítase al Anexo C.

	[8]	[69]	[46]	[61]	[29]	[83]	[84]	[24]	[66]	[1]
Perfil Contextual	+	?	-	-	-	-	-	-	?	+
Perfil de usuario	+	+	-	-	-	+	+	+	+	+
Registro de cambios	?	-	-	-	-	-	?	-	+	?
Semántica	?	+	+	+	+	+	?	+	?	?
Términos relacionados	+	+	+	+	+	+	?	+	-	-
Consultas históricas	-	-	+	?	?	+	-	+	-	-
Filtros	-	-	-	-	-	-	-	-	+	+
Uso de ontologías	?	-	?	-	+	+	?	-	?	+

Tabla 5. Comparación trabajos relacionados enriquecimiento.

Como se puede ver en la Tabla 5, la mayoría de los trabajos basan la adaptación de las consultas en un enriquecimiento semántico, en su gran mayoría utilizando diccionarios de sinónimos y términos relacionados; otros pocos trabajos utilizan un perfil de usuario para poder entregar resultados acordes a los gustos del usuario. Sin embargo, la cantidad de trabajos en los que se considera el uso de un perfil contextual es pequeña, y como se había mencionado anteriormente, el uso de un perfil de usuario y un perfil contextual son aspectos fundamentales para la adaptación del contenido. Por otra parte, se cuenta con el uso de consultas históricas lo que se

ve en la mayoría de los trabajos; estas son utilizadas en su gran mayoría para relacionar términos a incluir en el momento de enriquecer una consulta.

4. Metodología

En la propuesta del presente trabajo de grado se planteó una metodología con el fin de cumplir con los objetivos propuestos. A continuación se presentan la metodología propuesta, junto con la metodología realizada.

4.1. Metodología propuesta

Para la realización del presente trabajo, se contó en primera instancia con una fase exploratoria, donde se identificó la línea temática de trabajo, la problemática a atacar, los objetivos del trabajo y las actividades que se deberían llevar a cabo para alcanzar los objetivos. Además de éstos, dentro de la metodología expuesta en la propuesta del trabajo de grado, se contemplaron varias actividades que se pueden resumir en tres fases a realizar posteriormente: investigación, desarrollo y prototipado (ver Figura 13)

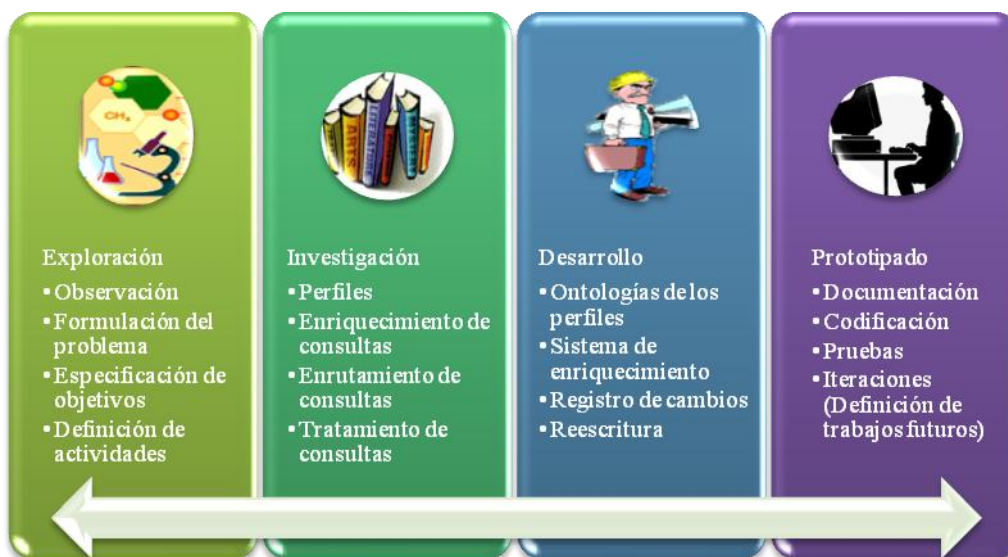


Figura 13. Metodología de la propuesta del trabajo de grado

- Fase de exploración: En ésta fase se buscó definir el área de investigación, la problemática a enfrentar, los objetivos y las actividades a realizar durante el trabajo, para alcanzarlos objetivos propuestos.
- Fase de investigación: Habiendo definido el problema a atacar y los objetivos, se definen las temáticas, esta fase consistía en recolección y análisis de de la información de dichas temáticas. Parte de esta fase, se desarrollo conjuntamente con la exploración, para identificar el problema y realizar el planteamiento del mismo.

- Fase de desarrollo: En esta fase se llevan a cabo las contribuciones, partiendo de la información recolectada y analizada en la fase anterior.
- Fase de implementación: A partir de las contribuciones, se procede a realizar los documentos necesarios para desarrollar el prototipo funcional, se implementa dicho prototipo y se realizan pruebas sobre éste. A partir de los resultados obtenidos en las pruebas y del trabajo realizado en las fases anteriores, se especifican las conclusiones y trabajos futuros.

4.2. Metodología realizada

Teniendo como base la metodología anterior, se comenzó el desarrollo del trabajo de grado. Se llevaron a cabo las cuatro fases presentadas (exploración, investigación, desarrollo y prototipado), la diferencia con la metodología propuesta, radica en que la fase de investigación se realizó en paralelo con las otras tres fases (ver Figura 14). Este cambio se presentó al ver la necesidad de investigar temas que no habían sido contemplados en la fase de investigación, mientras se desarrollaban las contribuciones y el prototipado, donde se continuó con la investigación, en menos intensidad, para el desarrollo tanto de los documentos, como para la codificación.



Figura 14. Metodología realizada

5. Contribuciones

En este capítulo se describirán los resultados obtenidos durante el desarrollo del presente trabajo de grado. Estos serán descritos de acuerdo a la metodología planteada en la propuesta: en la primera parte se explicará brevemente cómo está constituido el sistema de enriquecimiento; luego se procede a explicar el módulo de enriquecimiento de consultas, y el proceso que se lleva a cabo dentro de éste. En la siguiente sección se procede a definir los diferentes perfiles utilizados para personalizar las búsquedas, así como los componentes que estos contemplan. A partir de los perfiles, se explican unas reglas que serán utilizadas para establecer qué

información se agregara a las consultas. A continuación se describe el proceso utilizado para la reescritura de la consulta y el sistema de enrutamiento de consultas que será usado para este trabajo. Y con el fin de permitir el dinamismo del contexto (por ejemplo, el cambio de la ubicación del usuario) se describe el *Registro* de cambios y la forma en que es utilizado. Para finalizar se muestra el caso de estudio y los escenarios de prueba para el sistema de enriquecimiento de consultas, junto con los resultados obtenidos en las pruebas; para esta parte se contará con un prototipo funcional.

5.1. Sistema de enriquecimiento de consultas

Con el fin de adaptar la información retornada al usuario, se definió un sistema de enriquecimiento de consultas (ver Figura 15). Para empezar, el usuario introduce una consulta en una aplicación que utiliza el sistema de enriquecimiento. Ésta consulta, será dirigida al módulo de enriquecimiento de consultas, donde se le agregarán algunos elementos con el fin de enriquecerla. Los elementos que se agregan están relacionados con los perfiles de sesión e histórico, y a su vez, cada uno de estos perfiles está compuesto de dos partes: *i)* el perfil de usuario y *ii)* el perfil de contexto. A continuación, la consulta se enriquece por medio de una serie de reglas establecidas, que dependen de la aplicación. La consulta enriquecida se envía entonces al módulo de enrutamiento, que se encargará de decidir cuáles serán las fuentes de información en capacidad de satisfacerla. Luego, la fuente de información seleccionada proveerá un conjunto de resultados, que serán enviados a la aplicación para ser presentados al usuario.

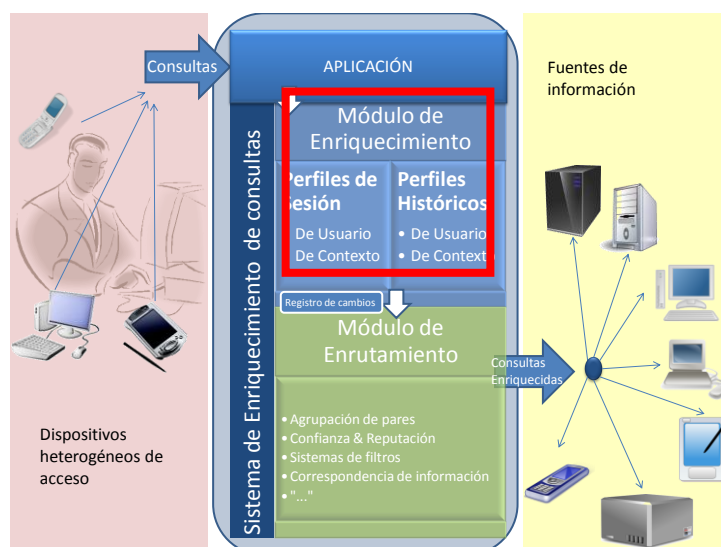


Figura 15. Sistema de enriquecimiento de consultas

El presente trabajo, se centra en el *Módulo de Enriquecimiento de Consultas* cuya estructura se muestra en la Figura 15, entre la aplicación y el módulo de enrutamiento. En éste se recibe la

consulta y se enriquece por medio de los perfiles del usuario. La consulta enriquecida se enviará al módulo de enrutamiento que se encarga de seleccionar un método de enrutamiento, el cual dirigirá la consulta enriquecida a los sistemas de información que la pueden responder y recibir los resultados que éstos envíen de vuelta. Como módulo de enrutamiento se utilizarán sistemas existentes que sepan sobre la organización de la información y qué SI puede responder total o parcialmente a las consultas, tratadas previamente, enviadas por el módulo de enriquecimiento.

En la siguiente sección, se muestra el proceso descrito anteriormente, por medio de un diagrama de actividad.

5.2. Esquematización del proceso de enriquecimiento

En la Figura 16, se presenta el proceso de enriquecimiento de consultas, cada parte del proceso será descrito con mayor profundidad en las siguientes secciones. En dicho proceso, primero el módulo de enriquecimiento recibe de la aplicación la consulta que el usuario ha ingresado. Luego se reciben los elementos que se deben tener en cuenta para el enriquecimiento y la información que dichos elementos manejan, esto para realizar la adaptación al contenido de acuerdo a las características del usuario. Luego de esto, se buscan las relaciones existentes entre los elementos de un mismo perfil y entre los perfiles, para así generar unas reglas que se encargan de definir la información que se agregará a la consulta. Dentro de las reglas definidas se buscarán aquellas que se puedan relacionar con la consulta original para realizar el enriquecimiento. Una vez escogidas las reglas, se debe hacer una reescritura para incluir las nuevas características. El módulo de enriquecimiento, enviará entonces al módulo de enrutamiento la consulta enriquecida para la búsqueda de resultados.

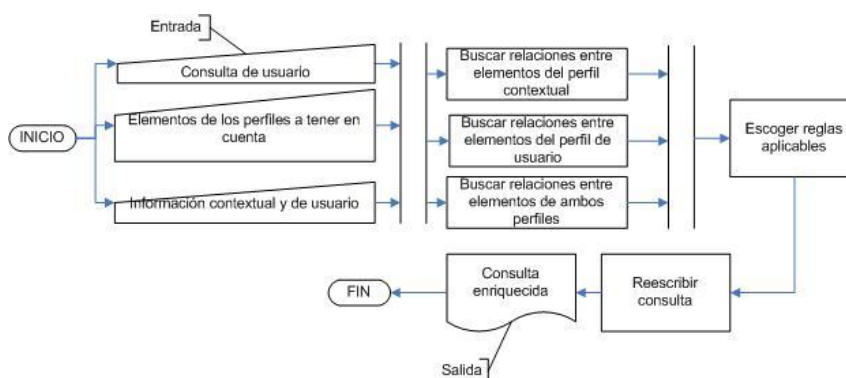


Figura 16. Diagrama del proceso de enriquecimiento de consultas

A continuación, se describe a profundidad el funcionamiento del módulo de enriquecimiento y el proceso utilizado para lograr el desarrollo de dicho módulo.

5.3. Módulo de enriquecimiento de consultas

El módulo de enriquecimiento es el encargado de agregar los parámetros utilizados para la adaptación de la consulta. Éste se muestra en la Figura 17, donde se pueden observar de manera general los componentes de adaptación involucrados: el modelo de adaptación, es el encargado de proveer los elementos relevantes de los perfiles de usuario (*MPU*) y de contexto (*MC*) que pueden ser aplicados para la sesión actual y que se convertirán en el perfil de dicha sesión (*PSA*). Además, se revisa el histórico del usuario en el sistema (*PsHs*) y se selecciona la información que pueda ser aplicada para la sesión actual. A esto se le conoce como (*PsHs*) filtrado. A la unión del *PSA* y el (*PsHs*) filtrado se le denomina “perfil de sesión” (*PS*) y es con estos elementos que se va a enriquecer la consulta Q que le llega al sistema de enriquecimiento de consultas; finalmente y de esta manera se genera Q' (consulta original enriquecida).

En la Figura 17, se muestran dos tipos de operaciones. La suma y la unión. La unión es la unión de conjuntos y la suma es una unión entre subconjuntos de los conjuntos especificados. Por ejemplo, el *PsHs* (Perfil histórico), es la suma de subconjuntos del Perfil Histórico del Perfil de Usuario y del Perfil Histórico del Perfil Contextual. Esto indica que no se necesita toda la información en ambos perfiles para poder llevar a cabo la construcción del perfil histórico. Mientras que, ya contruidos el *PSA* y el *PsHs*, si se necesita toda la información disponible en ellos para poder generar el perfil de sesión. Una parte de la información consignada en éste perfil de sesión será añadida a la información en la consulta del usuario.

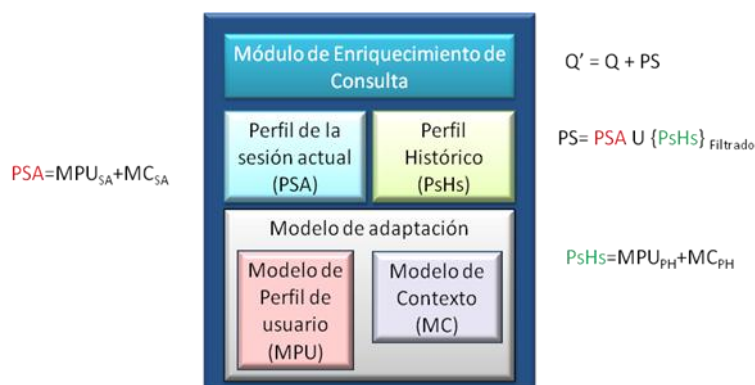


Figura 17. Funcionamiento del Módulo de Enriquecimiento de Consulta y generación de la consulta enriquecida a partir del perfil de sesión y la consulta original.

Teniendo claro el funcionamiento de alto nivel del sistema de enriquecimiento de consultas, se procederá a explicar paso por paso, el proceso que se lleva a cabo para la expansión de las consultas, dependiendo de los gustos, preferencias y el contexto de un usuario. En la Figura 18, se muestra el proceso seguido para el desarrollo del presente trabajo.



Figura 18. Proceso de desarrollo del sistema de enriquecimiento.

En las siguientes secciones, se procede a explicar cada uno de los pasos contemplados para el desarrollo del sistema de enriquecimiento de consultas. Comenzando con la definición de los perfiles de adaptación (usuario y contexto en su orden). Luego se hace la definición de las reglas (qué son y cómo se usan en el módulo de enriquecimiento). Después se definen los perfiles de sesión e histórico. Seguidamente se establece la priorización que se va a aplicar a las reglas con el fin de llevar a cabo el enriquecimiento de las consultas. El siguiente paso es la definición del proceso a seguir para reescribir las consultas, partiendo de la consulta ingresada por el usuario y añadiéndole la información que se haya consignada en los perfiles. Por último se define cómo incorporar al sistema de enriquecimiento el dinamismo del contexto, y cómo se actúa cuando se presenten cambios en el contexto.

5.3.1. Definición del perfil de usuario

En la Figura 19 se resalta el primer paso en el proceso de éste trabajo de grado: la definición del perfil de usuario, que se realizará a continuación.



Figura 19. Perfil contextual dentro de la proceso

Siendo los perfiles de usuario y contexto la base para la adaptación dentro del módulo de enriquecimiento de consultas, es importante notar que los componentes propuestos para estos perfiles pueden ser modificados (eliminándolos, cambiándolos o añadiendo nuevos componentes), para que así se puedan considerar las características de los usuarios y del contexto, que sean necesarias para la aplicación que está utilizando el sistema de enriquecimiento. Se puede realizar una personalización contemplando menos elementos de los propuestos dentro de los perfiles (es decir, más general), o, por el contrario, se puede una personalización más detallada añadiendo nuevos elementos a dichos perfiles.

Para el modelo de perfil de usuario, propuesto en el presente trabajo, se realizó una adaptación del *MAIPU* (Modelo de Adaptación de Información basado en Perfil de Usuario) para más información sobre *MAIPU* remítase al Anexo D y de *MAICO* (Modelo de adaptación de comunidades), el cual se presenta en el Anexo F. Estos proyectos fueron desarrollados conjuntamente, en el marco de un proyecto especial, realizado por estudiantes de la Pontificia Universidad Javeriana con miras a desarrollar un modelo de adaptación que integra llamado *I AM*, el cual integra varias dimensiones, como son: contexto, perfil de usuario, dispositivo de acceso y conexión inalámbrica.

El perfil de usuario presentado por *MAIPU* busca identificar al usuario dentro del sistema. Además se contemplan los gustos, intereses y preferencias del usuario.

Por otra parte, *MAICO* se enfoca en modelar comunidades. Este perfil cuenta con diversos módulos que permiten modelar una comunidad, los roles y las normatividad con que se rige un grupo de personas. Algunos de los componentes de *MAICO* son útiles dentro del perfil de

usuario, dado que permite modelar algunos comportamientos, a los cuales se debería someter las personas al ser miembros de una comunidad específica.

Teniendo en cuenta estos datos, así como lo consignado en el estado del arte, se determinó que los componentes a incluir dentro del perfil de usuario son: *i*) preferencias de actividad, *ii*) preferencias de actividades relacionadas, *iii*) información básica, *iv*) gustos, *v*) intereses, *vi*) roles y *vii*) normas. La representación visual del perfil de usuario, se puede encontrar en la Figura 20.



Figura 20. Perfil de usuario propuesto

A continuación se describen de manera resumida, los componentes que conforman el perfil de usuario utilizado en el presente trabajo.

5.3.1.1. Componentes del perfil de usuario

MAIPU define de la siguiente manera cada uno de sus componentes²⁰:

Preferencias de usuario: Es un módulo que describe las preferencias que un usuario tiene respecto a las actividades que realiza. Los sub componentes que posee son los siguientes:

- Preferencia de actividades relacionadas: De acuerdo a las actividades realizadas por el usuario, éstas pueden estar conformadas por otras actividades que apoyan el proceso que permite llevar a cabo la actividad principal.

-
- ²⁰ Resumen del documento donde se describe *MAIPU*, Anexo D
 - Anexo C. Trabajos relacionados con el perfil de usuario, perfil de contexto, la reescritura de consultas y el enriquecimiento de consultas
 - Anexo D

- Gustos: Se definen teniendo en cuenta las cinco dimensiones que hacen referencia a los sentidos. Estas dimensiones le permiten al hombre conocer y relacionarse con su entorno, describiendo entonces, aquello que estimula los sentidos del usuario.
- Intereses: Primero que todo vale la pena aclarar que los intereses cambian constantemente. Esto es un reflejo de la evolución de los usuarios en su disposición de adquirir productos/servicios, exigiendo cada vez mayor calidad. Entre los factores que pueden afectar se encuentran los cambios climáticos, las estaciones, la edad, el entorno en el que vive, entre otros.
- Información básica: Es la información de un usuario que no cambia frecuentemente y que lo identifica en el sistema. Entre las características se pueden mencionar el nombre, el apellido, el email y la fecha de nacimiento.

MAICO define de la siguiente manera cada el siguiente componente²¹:

- Roles: Son grupos conformados por uno o más usuarios; estos usuarios tienen ciertos privilegios, es decir, tienen permiso para realizar ciertas actividades, y así mismo existen otras que no se les permite realizar. Lo anterior sirve para controlar el comportamiento de los usuarios pertenecientes a una comunidad.

5.3.1.2. Representación de los elementos del perfil de usuario

Para la representación de este modelo se utilizaron ontologías. Una ontología es una representación de las propiedades de un elemento²². Según esto, por medio de éstas se pueden realizar representaciones (entre otros), de los componentes incluidos en los perfiles, describiendo al mismo tiempo los subcomponentes que lo conforman. La estructura de las ontologías hace que sean de fácil entendimiento. Esto se debe a que se parte, de un elemento general, que se va desglosando para llegar así llegar a los subelemento [65]. En la Figura 21 se muestra la representación ontológica del perfil de usuario.

• ²¹ Resumen del documento donde se describe *MAICO*, Anexo E

Anexo E.

²² www.rae.es



Figura 21. Ontología del perfil de usuario

A continuación se procede a especificar las relaciones que se presentan entre los componentes del perfil de usuario propuesto en el presente trabajo.

5.3.1.3. Comunicación entre los componentes del perfil



Figura 22. Comunicación entre componentes del perfil de usuario

El módulo de preferencias (tal como se ve en la Figura 22), está construido a partir de los gustos e intereses que tiene un usuario. Para más información acerca de la representación ontológica, la adquisición de información y la instanciación del perfil, remítase al Anexo G.

En la siguiente sesión se procede a explicar los componentes que describen la interacción entre el usuario y la aplicación.

5.3.2. Perfil contextual

Como se puede ver en la Figura 23, para el proceso de realización del *Sistema de Enriquecimiento de Consultas*, se desarrolló en segundo lugar el perfil de contexto, que es un complemento al perfil de usuario para el proceso de enriquecimiento. En ésta sección se definirá el perfil de contexto, sus componentes y la descripción de los mismos.

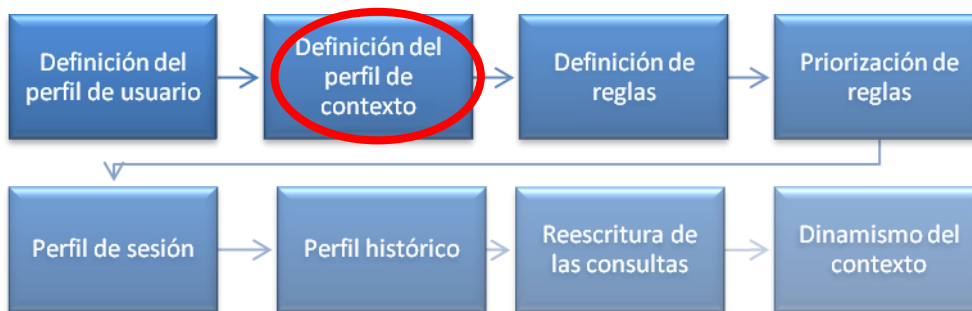


Figura 23. Perfil contextual dentro de la proceso

Para la definición del perfil contextual, se utilizó un modelo desarrollado en el marco del presente trabajo de grado, conjuntamente con otro modelo de perfil contextual definido para un modelo de adaptación general llamado *I AM* (*Modelo Adaptable Integrado* por sus siglas en inglés), desarrollado en un proyecto especial por un grupo de estudiantes de la Pontificia Universidad Javeriana.

Antes de explicar el perfil contextual, es necesario tener claro qué es el contexto en sí. La definición que se utilizara en el presente trabajo, parte de la explicación dada Dey *et al.* [26], se expone que el contexto es el conjunto de los elementos utilizados para representar la interacción entre el usuario y el sistema.

En la Figura 24 se presenta el perfil contextual propuesto para el presente trabajo de grado, el cual es una adaptación del módulo de contexto de *I AM*²³.

El modelo de contexto MOCA (acrónimo de MOdelo Contextual de Adaptación, Figura 24), propuesto para este trabajo contempla características espacio/temporales, ambientales, de actividad, y un módulo donde se encuentra el identificador del usuario con el que está relacionado el perfil contextual, con lo cual se puede determinar con qué usuario está relacionado el perfil contextual.

²³ Características contextuales: aquellas que describen la interacción del usuario con el sistema.

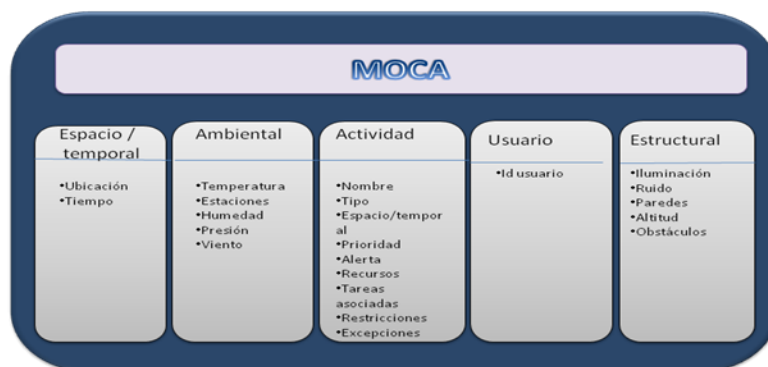


Figura 24. Perfil contextual utilizado en el módulo de enriquecimiento

A continuación se describirán los componentes que se muestran en la Figura 24.

5.3.2.1. Componentes del perfil contextual

Si bien en este trabajo de grado se van a usar los componentes contextuales listados en el párrafo anterior, el modelo permite añadir otros componentes contextuales que no hayan sido contemplados o eliminar componentes que no sean necesarios dependiendo de las necesidades de la aplicación (tal como se especifica para el perfil de usuario). Esto se realiza con el fin de proveer un enriquecimiento de consultas a aplicaciones complejas, que contemplen más elementos del contexto.

Para cumplir con el objetivo de enriquecer las consultas provistas por un usuario, se considera necesario saber quién es ese usuario. Por esto se ha contemplado en el modelo una relación entre perfiles para asociar un contexto a un usuario. El componente espacio/temporal contempla la ubicación del usuario y la hora actual. En el componente ambiental se modelan aquellas características particulares del ambiente en el que se desenvuelve la interacción entre el usuario y el sistema. Estas características pueden influir en las preferencias de los usuarios dados cambios, por ejemplo, en la temperatura o en estaciones del año entre otros. Las actividades definen lo que el usuario hace en un momento determinado, y pueden influir en la aplicación de sus preferencias.

Los componentes que son tenidos en cuenta para el perfil de contexto del presente trabajo de grado son:

- **Espacio/Temporal:** Módulo que describe la ubicación de las personas y un momento determinado durante el día en el que se lleva a cabo la interacción del usuario con el sistema. Éste se subdivide en Ubicación y tiempo.
 - **Ubicación:** Indica la localización del usuario.

- Tiempo: Es definido como el momento en el que el usuario interactúa con el sistema. El tiempo completo se especifica como: “día, mes, año, hora, minutos, segundos”. También se incluye un periodo que indica un lapso de tiempo y una periodicidad, que indica cada cuánto se repite una situación (por ejemplo, se puede definir el tiempo que pasa entre dos realizaciones de la misma actividad).
- Ambiental: Define las características no físicas del lugar en que se encuentra el usuario, que pueden afectar las preferencias y/o las necesidades de los usuarios. Está compuesto por los componentes que se especifican a continuación.
 - Temperatura: Indica la temperatura existente en el ambiente en el momento en el que el usuario interactúa con el sistema. La diferencia en la temperatura influye en las necesidades y preferencias del usuario. Por ejemplo, con frío una persona puede preferir un chocolate caliente, mientras que con calor puede preferir un helado.
 - Estación: Las diferentes estaciones influyen en las actividades que se realizan. Por ejemplo, en verano un usuario determinado planea ir a comer helado, mientras que en invierno este mismo planea ir a tomar chocolate caliente. Para las zonas en donde hay estaciones, es decir, por fuera de la zona tórrida, los valores posibles para las estaciones serán: otoño, invierno, primavera y verano. Para los países que están en la zona tórrida, es decir, aquellos que no tienen este mismo tipo de estaciones, los valores posibles serán: Estación seca y estación lluviosa.
- Actividad: En el contexto es importante tener en cuenta las actividades que van a realizar los usuarios. Esto para que se puedan establecer restricciones como no recibir mensajes o llamadas durante una reunión (es decir, no recibir cierta información en cierto momento). A su vez se pueden definir excepciones a dichas restricciones. Tomando el ejemplo anterior se puede definir una excepción como poder recibir llamadas de un jefe o un familiar cercano. También se define dónde se va a realizar la actividad, los recursos y tareas necesarias para llevarla a cabo, y un recordatorio que es usado si el usuario desea recibir una notificación para recordar el inicio próximo de una actividad. También se define una prioridad que define cuál de las actividades es más importante en el caso en que dos o más se traslapen.
- Usuario: Cada perfil contextual está asociado a un usuario específico, el cual a su vez contiene un perfil de usuario. Esta relación se hace mediante un identificador. El identificador que se usará en el desarrollo de éste trabajo de grado será el correo electrónico del usuario.

- **Estructural:** El módulo estructural incluye una descripción física del lugar en el cual se encuentra el usuario. Este módulo agrega valor a las aplicaciones que necesiten conexiones inalámbricas, dado que los componentes estructurales alrededor del dispositivo pueden limitar la capacidad de conexión de dichos dispositivos o incluso, pueden afectar el desempeño de las conexiones. Este componente también es relevante en aquellas ocasiones en las cuales las características estructurales tales como la luminosidad y el ruido pueden afectar las tareas o los resultados de una aplicación (por ejemplo, aplicaciones sensibles a la luz o resultados en formato audio). El módulo estructural está compuesto por los siguientes componentes.
 - Luminosidad²⁴: es definida como la magnitud física emitida por una fuente de luz en ciertas condiciones (sol, bombillos, entre otros). Este elemento podría influir en el momento de desplegar información, debido a que al contar una mayor luminosidad en el entorno, la pantalla *LCD* de los dispositivos de acceso pueden reflejar objetos y dificultar la visualización de la información en la pantalla.
 - Ruido: Son sonidos no articulados, que por lo general interfieren para que el usuario pueda interpretar la información entregada en formato de audio. Si el usuario se encuentra en un lugar con mucho ruido sería necesario subir el volumen y en algunos casos se podría denegar el servicio (llamadas, contenido multimedia), o así mismo se podría disminuir el volumen si la condición indica que existe poco ruido.
 - Piso, altitud, obstáculos: el piso indica en qué nivel de un edificio se encuentra el usuario, mientras que la altitud hace referencia a la altura respecto al nivel del mar. Los obstáculos se refieren a la cantidad de interferencia que puede haber, por ejemplo 10 árboles, 2 paredes. Dependiendo del valor de estos subcomponentes la intensidad de las señales inalámbricas pueden variar.

5.3.2.2. Representación de los elementos del perfil de contexto

Para la representación de este modelo se utilizarán ontologías, las razones por las cuales se usan ésta representación son las mismas descritas para el perfil de usuario. En la Figura 25, se presenta la representación ontología del perfil contextual.

²⁴ Adaptado de la Real Academia de la Lengua Española

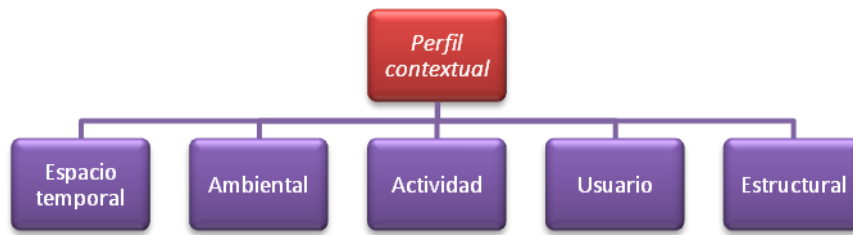


Figura 25. Representación ontológica del perfil de contexto²⁵

5.3.2.3. Comunicación entre los componentes del perfil contextual

Dado que algunos de los componentes del perfil contextual se pueden definir en relación con otros componentes, en esta sección se presenta la interacción entre ellos.

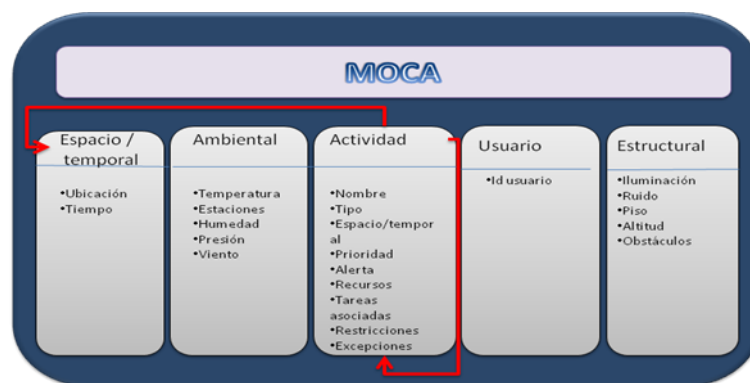


Figura 26. Interacción entre componentes contextuales

En la Figura 26, se puede observar que una actividad necesita del módulo espacio/temporal. Incluso, uno de los subcomponentes del componente actividad es como tal el componente espacio/temporal. Esto se debe a que una actividad cuenta con un lugar de ejecución, así como con un momento del día en el que se lleva a cabo.

También se observa una relación recursiva de actividad. Esta relación se debe a que una actividad se define como un conjunto de tareas y en algunos casos de otras actividades.

Si desea obtener más información acerca del perfil de contexto, remítase al Anexo F en donde se encontrará la representación, instanciación y adquisición de información de este perfil.

5.3.3. Comunicación entre el perfil de usuario y el perfil de contexto

Dado que el fin último del enriquecimiento de consultas, es poder brindar al usuario información adaptada, dependiendo de quién es (perfil de usuario) y del entorno en el que se

²⁵ Para más detalles sobre el perfil contextual y su representación remítase al Anexo F

encuentra al interactúa con la aplicación (perfil contextual), se debe lograr una comunicación entre estos dos perfiles.

En esta sección se describen los componentes que interactúan entre los dos perfiles y el formato en el que se efectúa la comunicación.

COMPONENTES RELACIONADOS			
Proveedor	Cliente	Datos a intercambiar	Formato ²⁶
Perfil de Usuario, componente de Información básica	Perfil Contextual , componente de usuario	Identificador de usuario	<identificador> <i>e.g.</i> <faragon@javeriana.edu.co>

Tabla 6. Perfil de usuario como proveedor, perfil contextual como cliente

En la Tabla 6 se muestra la interacción existente entre los dos perfiles (el perfil de usuario provee la información y el perfil contextual la utiliza).

La Tabla 7 muestra la información que el perfil contextual envía al perfil de usuario (desde el módulo contextual de actividad hacia el módulo del perfil de usuario de preferencia de actividad relacionada).

COMPONENTES RELACIONADOS			
Proveedor	Cliente	Datos a intercambiar	Formato
Perfil contextual , Actividad	Perfil de usuario , Preferencia de actividad relacionada	Nombre de la actividad, Espacio/temporal, duración, recursos, restricciones, tareas, alertas, restricciones, excepciones	(Nombre, { espacio/temporal}, duración recursos, restricciones, <tareas>, <restricciones>, <excepciones>)

Tabla 7. Perfil contextual como proveedor, perfil de usuario como cliente

El fin de los datos consignados en las dos tablas anteriores, es poder ver cuáles son las relaciones existentes entre los componentes del perfil contextual y del perfil de usuario. A su vez se especifican los datos que necesita el perfil cliente del perfil proveedor para el correcto funcionamiento del sistema, así como el formato de transmisión para que el cliente pueda entender los datos que le entrega el proveedor.

Luego de explicar cómo están conformados los perfiles, su representación y la comunicación que existe entre ellos, se procede a explicar cómo se puede hacer la adición de información a los perfiles de contexto y de usuario. Esto con el fin de hacer modificable el modelo de adaptación (para incluir más o menos preferencias según el tipo de aplicación que se vaya a utilizar).

²⁶ Para más información sobre los formatos para el paso de información remítase al ANEXO MAIPU, cuando se deseen realizar nuevas relaciones se debe utilizar dicho formato.

5.3.4. Cambios a los perfiles

Como se ha dicho con anterioridad, los perfiles presentados en el documento, son un caso particular de los mismos, y dependiendo de la aplicación estos podrían presentar modificaciones, por ejemplo se podrían añadir, eliminar o modificar los componentes de los perfiles propuestos.

La adición de componentes a un perfil se puede hacer sin mayores traumatismos, en la medida en que la información que se añade mantenga la integridad referencial de las relaciones entre los perfiles y entre los componentes del mismo perfil. Es decir, si se modifica el componente de información básica de usuario, entonces se deberá modificar el módulo de usuario (perfil contextual) con el fin de reflejar este cambio. Si se desea modificar un componente en cualquiera de los dos perfiles, se debe seguir los pasos enunciados para añadir, con la diferencia que no se agregará un componente completo, pero si se cambiarán los elementos de dicho componente. Es por esta razón que se deben revisar las relaciones existentes, las nuevas relaciones y el contenido que manejará cada nuevo elemento. Por ejemplo, si la aplicación que utiliza el sistema de enriquecimiento de consultas considera que el componente de comunidad no es útil para el fin de la misma, este podría ser eliminado o modificado para que se adapte a las necesidades de la aplicación,

En el caso de la eliminación de elementos del perfil, se deberá primero modificar aquellos elementos que estén relacionados con el componente que se va a eliminar. Por ejemplo, en el caso de eliminar el componente espacio/temporal, se debe a su vez, modificar los componentes actividad (Perfil de contexto), preferencias de actividad (perfil de usuario) y preferencias de actividad relacionada (Perfil de usuario). Estas modificaciones deben hacerse para que al eliminar uno de los componentes no queden los demás en un estado inconsistente debido a que utilizaban el componente a ser eliminado.

En la Figura 27 se muestra el proceso que se sugiere seguir para realizar un cambio a los perfiles. Cabe aclarar que este proceso no lo realiza el sistema de enriquecimiento de consultas, sino que es el diseñador de la aplicación el encargado de verificar la pertinencia, y los cambios consecuentes de realizar alguna modificación a los perfiles.

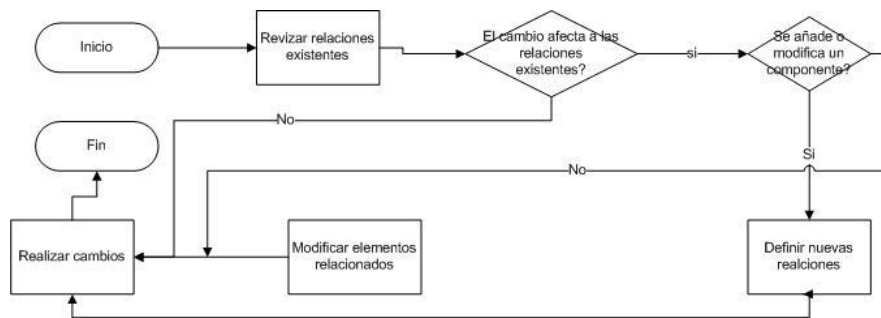


Figura 27. Proceso para realizar cambios en los perfiles

5.3.5.Reglas

En la Figura 28, se muestra resaltado el tercer paso del proceso que lleva a la definición del módulo de enriquecimiento. En esta sección se detallará la definición de regla y cuál es su papel que tiene dentro del módulo de enriquecimiento de consultas.



Figura 28. Definición de reglas dentro del proceso

Una regla define qué hará el módulo de enriquecimiento cuando se cumple una o más condiciones, siendo así, una definición lógica del tipo $(X \rightarrow Y)$ ²⁷. Existen dos tipos de reglas que se pueden definir en el módulo de enriquecimiento de consultas: *i)* las reglas de equivalencia y *ii)* las reglas de componente. Las primeras, relacionan conceptos²⁸ o valores de los mismos entre sí, tomando como base un diccionario de equivalencias. El segundo tipo relaciona componentes de los perfiles, conceptos y valores. A continuación se procede a describir dichas reglas, para qué sirven y cuál es la forma en que se aplican.

²⁷ Si la condición X se cumple entonces se realizará la acción Y.

²⁸ En el ámbito de este trabajo, un concepto es definido como una forma de representar un elemento l y, un valor será una instancia de esa expresión. Por ejemplo, el concepto "distancia", puede tener como valor "500 metros".

En la Figura 29, se muestra cómo se define una regla. En la parte izquierda se define la condición a cumplir y en la derecha el proceso a ejecutar, es decir, el conjunto de operaciones o acciones que se llevan a cabo sólo si la condición es verdadera. Las condiciones pueden definirse respecto a la consulta que el usuario ingresa o respecto a los componentes de los perfiles.

Condición → Proceso
Figura 29. Definición de reglas

El proceso se realizará sobre una consulta²⁹ provista por un usuario. Las acciones posibles dentro del proceso, pueden ser la adición, reemplazo o exclusión de términos³⁰ en la consulta, con el fin de reflejar el enriquecimiento semántico propuesto en este trabajo de grado. Los procesos que hacen estos dos tipos de operaciones sobre la consulta del usuario y su funcionamiento serán explicados posteriormente en la sección 5.3.9.

5.3.5.1. Reglas de equivalencia

Las reglas de equivalencia, son usadas para que el usuario y la aplicación manejen terminología homogénea, logrando así, que cuando se utilicen conceptos diferentes a los utilizados por la aplicación, pero que tengan el mismo significado, estos puedan unificarse. Mediante estas reglas se puede generar un diccionario de sinónimos.

Antes de definir las reglas de equivalencia, se deben establecer relaciones de equivalencia, que puede ser entre conceptos, o entre un concepto y sus posibles valores.

En primer lugar, una relación de equivalencia se puede establecer cuando un usuario ingrese su concepto, con el fin de unificarlo con el concepto manejado en la aplicación (en caso de que sea diferente). Es decir, el usuario puede tener un concepto de distancia, mientras que la aplicación puede manejar el concepto de localización. La relación de equivalencia, estaría dada por los conceptos de distancia y localización, como sinónimos. A su vez, también se pueden establecer rangos con respecto a los cuales hacer una búsqueda, por ejemplo, reemplazando la palabra *cerca* por un rango entre 0 y 500 metros.

En segundo lugar, las relaciones pueden establecerse entre los valores que podrían tomar los conceptos. Por ejemplo, el valor *cerca* para el usuario puede simbolizar el valor *500 metros*. Cabe destacar que aunque los conceptos pueden ser diferentes, los valores tanto para el usuario como para la aplicación pueden ser los mismos. Por ejemplo, una relación de equivalencia

²⁹ Esta consulta es un conjunto de palabras, que el usuario o la aplicación envían al módulo de enriquecimiento. Es similar a los conjuntos de términos que se usan para realizar consultas en motores de búsqueda al estilo de Google®.

³⁰ Un término puede ser una palabra o una frase nominal.

puede estar definida entre los conceptos localización y distancia, y ambos podían ser manejados en escala de metros.

Las relaciones de equivalencia entre valores pueden ser definidas como un rango o un valor específico. En la Tabla 8, se muestra en las tres primeras filas un ejemplo de equivalencia por rangos, mientras que en la última fila, una equivalencia de un valor específico.

Valor del usuario	Valor del sistema
Cerca	< 501 metros
Intermedio	Entre 501 metros y 1000 metros
Lejos	> 1000 metros
Universidad	Carrera 7 # 40-62

Tabla 8. Relaciones de equivalencia

Usando estas relaciones se pueden establecer cuatro escenarios (ver Tabla 9), en los que la definición de reglas de equivalencia puede presentarse: *i*) cuando el concepto que tienen el usuario y la aplicación es el mismo y los valores que puede tomar son iguales (*e.g.* usuario y aplicación manejan el concepto *hora*, y ambos pueden trabajar con el mismo formato). En este escenario, no es necesario definir relaciones de equivalencia. *ii*) Tanto usuario como aplicación manejan el mismo concepto pero los valores que puede tomar son diferentes (*e.g.* ante el concepto *fecha*, el usuario puede considerar valores con el formato *AAAA/MM/DD*, y la aplicación puede considerar que la fecha se escribirá como *DD/MM/AA*). *iii*) Existe una diferencia entre los conceptos pero los valores pueden ser los mismos (*e.g.* El usuario maneja el concepto de *localización* pero la aplicación maneja *ubicación*. Sin embargo un valor para ambos conceptos puede ser *Bogotá*). *iv*) el último caso contempla el hecho en el cual el usuario y la aplicación manejan conceptos y valores distintos (*e.g.* el usuario tiene el concepto *ubicación* y la aplicación maneja *localización*. Los valores que toman son respectivamente “*Bogotá*”, y “*Latitud Norte: 4° 35'56*”, *Longitud Oeste: 74°04'5*”³¹ Para los tres últimos escenarios se puede definir relaciones de equivalencia.

³¹ Localización geodésica de Bogotá, <<http://www.bogota.gov.co/portel/libreria/php/01.270701.html>>, última consultado septiembre de 2009

Escenario	Concepto	Formato (valores que puede tomar el concepto)	¿Se sugiere definir regla de equivalencia?
1	Concepto aplicación = Concepto usuario	Formato aplicación = Formato usuario	No
2	Concepto aplicación ≠ Concepto usuario	Formato aplicación = Formato usuario	Si
3	Concepto aplicación = Concepto usuario	Formato aplicación ≠ Formato usuario	Si
4	Concepto aplicación ≠ Concepto usuario	Formato aplicación ≠ Formato usuario	Si

Tabla 9. Posibles escenarios para las relaciones de equivalencia

A partir de las relaciones de equivalencia se generan unas reglas, en las cuales se define qué hacer con dichas relaciones, si un concepto previamente establecido, se encuentra dentro de la consulta ingresada a la aplicación. De esta forma, en una regla de equivalencia como la que se muestra en la Figura 30, se evalúa si el concepto1 está presente en la consulta del usuario, para que luego se ejecute un proceso.

(Consulta.contiene("concepto1")) → Proceso
Figura 30. Generalización de las reglas de equivalencia

El proceso consiste en una serie de acciones que afectarán la consulta original (Q). Éste puede incluir la adición, el remplazo, o la exclusión de conceptos en la consulta. Los ejemplos de estas acciones se muestran a continuación.

La primera acción es agregar conceptos a la consulta Q . La adición de términos se usa para complementar y/o contextualizar la consulta del usuario. En la Figura 31 se muestra el esquema de cómo será una regla de adición de conceptos a la consulta del usuario.

(Consulta.contiene("concepto1")) → adicionar ("concepto2")
Figura 31. Esquema de adición de términos

Por ejemplo, se puede definir que si la consulta se hace sobre restaurantes, se adicionará el nombre de la ciudad en donde se encuentra el usuario, para así restringir la cantidad de resultados entregados (ver Figura 32).

(Consulta.contiene("restaurante")) → adicionar("Bogotá")
Figura 32. Ejemplo de adición de términos

La segunda acción posible es el remplazo de un concepto o un valor por otro. Con esto se busca poder cambiar conceptos o valores brindados por el usuario, por otros que la aplicación maneja como sinónimos. En el caso en el que un concepto del usuario no tenga un sinónimo manejado por el sistema, dicho concepto se mantendrá igual, y así se enviará al enrutador de consultas. En la Figura 33 se muestra el esquema para definir una regla de remplazo. En el

proceso, se puede reemplazar concepto que se evaluó en la condición (concepto1), u otro concepto diferente (en la Figura 33, el concepto2).

(Consulta.contiene("concepto1")) → reemplazar("concepto2", "concepto3")

Figura 33. Esquema de reemplazo de términos

Un ejemplo de cuándo aplicar un reemplazo de concepto es el siguiente: el usuario ingresa una consulta que contiene el concepto "ubicación", éste se reemplaza por "localización", que es el que la aplicación utiliza. La Figura 34 muestra cómo se establece esta regla, a partir del esquema de reemplazo de términos.

(Consulta.contiene("localización")) → reemplazar("localización", "ubicación")

Figura 34. Ejemplo de reemplazo de términos

Por último, la tercera acción que se puede realizar sobre la consulta provista por el usuario será excluir un concepto. Esta acción consistirá en descartar los resultados que contengan el concepto excluido. De esta manera se podría implementar un control de acceso y de contenidos. En la Figura 35 se muestra el esquema de una regla de éste tipo.

((Consulta.contiene("concepto1")) → excluir ("Concepto2"))

Figura 35. Esquema de exclusión de términos

Un posible ejemplo para la exclusión de conceptos es: al ingresar *conflicto* se podría evitar los resultados que contengan la combinación "conflicto armado", pero mostrar los que tengan la palabra original (*conflicto*). La forma de hacer esto es excluir el término *armado*, por medio de una regla de equivalencia. En la Figura 36 se muestra dicho ejemplo.

((Consulta.contiene("concepto1")) → excluir ("Concepto2"))

Figura 36. Ejemplo de exclusión de términos

Habiendo explicado las reglas de equivalencia, se procederá a definir las reglas de componente.

5.3.5.2. Reglas de componente

El proceso de enriquecimiento de consultas realizado a partir de las reglas de equivalencia puede complementarse por medio de las reglas de componente. Las primeras definen concordancias entre conceptos, a fin de establecer similitudes semánticas para la aplicación, mientras que las segundas se basan en relaciones entre los componentes de un mismo perfil y entre los componentes de diferentes perfiles.

Las relaciones de componente se pueden establecer entre un componente, el valor que podría tomar y un concepto específico, tal como se muestra en la Tabla 10. En dicha tabla se muestran las relaciones para dos componentes del perfil contextual que podrían denominarse “sencillas”.

Perfil	Componente	Valor	Concepto
Contexto	Espacio/Temporal.localización	Universidad	Académico
Contexto	Actividad	Estudio	Java
Contexto	Espacio/Temporal.tiempo.hora && Actividad.nombre	14:00 && Ocio	Café

Tabla 10. Relaciones de componente sencillas

Asimismo, se pueden establecer relaciones entre componentes del mismo perfil o entre componentes de diferentes perfiles que se denominarían relaciones complejas. En las dos primeras filas de la Tabla 11, se puede ver cómo son las relaciones de un mismo perfil y en la última fila, cómo es una relación entre componentes de diferentes perfiles.

Perfil	Componente	Perfil relacionado	Componente relacionado
Usuario	Gusto	Usuario	PreferenciaDeActividad
Contexto	Actividad.nombre	Contexto	Actividad.localización
Usuario	InformaciónBásica	Contexto	Gustos
Contexto	Actividad	Usuario	Gusto

Tabla 11. Ejemplo relaciones de componente complejas

A partir de las relaciones anteriores se pueden establecer las reglas de componente; éstas pueden definirse para un solo componente, verificando su valor en la condición y realizando un proceso³² si se cumple la condición. Por ejemplo, en el caso en el que el usuario esté en la universidad, se harán búsquedas relacionadas con temas académicos. En la Figura 37 se muestra cómo se implementa dicha regla.

(Contexto.Actividad.localización == “Universidad”)→adicionar (“Académico”)
 Figura 37. Ejemplo de regla valor-componente (sencillas)

Además, una regla de componente puede considerar más de un componente. En este caso, en la condición de la regla se consideran varios componentes usando operadores lógicos. Un ejemplo de esto es la regla definida en la Figura 38. En este ejemplo, se relacionan los componentes Localización y Nombre de la actividad, para establecer la regla: “Si la localización es la Universidad y la actividad es trabajo, entonces se debe adicionar a la consulta el término “Académico”.

³² Cabe aclarar que este proceso puede incluir acciones como: añadir, reemplazar o excluir. Al igual que en las reglas de equivalencia.

(Contexto.Actividad.localización=="Universidad" && Contexto-
Actividad.nombre=="Trabajo")→ adicionar("Académico")

Figura 38. Ejemplo de regla compuesta de valor-componente

También se pueden definir reglas en las cuales el proceso involucre el valor que posee un componente(a partir de relaciones como las de la Tabla 11). Un ejemplo de esto se encuentra en la Figura 39; en dicho ejemplo se adiciona la localización de la actividad ("Universidad Javeriana"); este ejemplo podría ser útil para una aplicación donde se muestren sitios de entretenimiento, restringiendo los resultados a los sitios que se encuentren cerca³³ al usuario.

(Contexto.Espacio/Temporal.tiempo.hora == "14:00")→
adicionar("Contexto.Actividad_Actual.localización")

(Contexto.Espacio/Temporal.tiempo.hora == "14:00")→ adicionar("Universidad Javeriana")

Figura 39. Ejemplo de regla de componente-componente (compleja) e instanciación

El segundo caso relaciona valores de componentes entre sí como la localización actual y la localización de la actividad o el tiempo actual y el lapso que dure una actividad. En la Figura 40 se muestra una regla que compara la localización actual y la localización de la actividad actual y luego adiciona el término "Bogotá".

(Contexto.Espacio/Temporal.localización == Contexto.Actividad_Actual.localización)→
adicionar("Bogotá")

Figura 40. Ejemplo de regla de componente comparando valores de componentes.

Otra posible utilidad de las reglas de componente sería el poder definir restricciones de acceso de acuerdo a los roles de un usuario. Estos se pueden usar para restringir el acceso a cierta información. Por ejemplo, un niño que busca algún tipo de información podría encontrarse con resultados relacionados que puedan distraerle o ser perjudiciales para él (por ejemplo, violencia o pornografía). Para esto se podrán suprimir esos resultados del conjunto final presentado al usuario con el rol "menor de edad" (ver Figura 41).

(Usuario.comunidad.rol == menor_de_edad) → excluir ("pornografía"), excluir("violencia" &&
"pornografía")

Figura 41. Ejemplo de regla de componente respecto a rol

A partir de las reglas de equivalencia y de componentes, se ejecuta un proceso para realizar la reescritura de consultas, dicho proceso se explicará en la sección 5.3.9.

³³ La cercanía es definida a partir de las reglas de equivalencia

5.3.5.3. Persistencia de las reglas

Luego de definir las reglas, es importante especificar cómo se van a almacenar, para que el módulo de enriquecimiento de consultas pueda usarlas. En esta sección se define el formato en que se hará el almacenamiento. En el caso de este sistema de enriquecimiento, los usuarios se conectarán a un servidor donde éste se aloja y allí se guardará la información para su correcto funcionamiento (reglas perfiles y consultas previas). En particular, para las reglas, centralizarlas en un mismo lugar, hará más fácil su edición (agregar nuevas, eliminar o editar reglas actuales), y su manejo (al cargarlas para llevar a cabo el proceso de enriquecimiento). Los usuarios tendrán un conjunto de reglas actualizadas cada vez que inicien una nueva sesión. Esta información se almacenará en un archivo con formato *XML*.

A la hora de definir el formato de almacenamiento es importante tener en cuenta ambas partes de la regla: condición y proceso, puesto que cada una tendrá una representación propia.

La manera en que se guardarán las relaciones de equivalencia se muestra en la Figura 42; el concepto de la izquierda será el relacionado con el usuario y el de la derecha el relacionado con la aplicación. Estos conceptos se pueden usar para representar una palabra, una frase, o un valor.

“Concepto1” = “Concepto2”

Figura 42. Esquema de una relación de equivalencia

5.3.5.3.1. Persistencia de las reglas de equivalencia

Cuando se estableció cómo se almacenan las relaciones de equivalencia, se definió que a la izquierda se pondría el concepto o el valor que el usuario maneja y a la derecha el concepto o valor con el cual se relaciona en la aplicación. Respecto a las reglas, a la izquierda se pondrá la condición de acuerdo a la consulta que ingrese el usuario. Luego, a la derecha el qué hacer cuando se cumple la condición y en la mitad, entre la condición y el proceso, se ubicará el símbolo >>> para indicar la transición entre estas dos partes de la regla.

Para guardar una regla de equivalencia, se tiene en la primera parte de la regla una condición aplicada a la consulta del usuario, la forma en que se realiza se muestra en la

Figura 43. La parte *consulta* hace referencia a aquella que provee el usuario y sobre la cual se está realizando el proceso de enriquecimiento.

Consulta.contiene(“Concepto”)

Figura 43. Esquema de condición para reglas de equivalencia

Para poder especificar cuál es la acción a realizar sobre la consulta Q , se dará el nombre de la acción y luego entre paréntesis el(los) concepto(s) que se utilizarán para el proceso indicado. En la Figura 44 se muestra el esquema de la adición.

adicionar("Concepto")
Figura 44. Esquema de escritura de adición

La acción de reemplazar que se muestra en la Figura 45; en esta se presentando dos conceptos, el primero que se encuentra dentro de la consulta Q y el segundo es el concepto por el cual se reemplazará.

reemplazar ("Concepto1,Concepto2")
Figura 45. Esquema de escritura de reemplazo

En la Figura 46, se muestra el caso particular de las reglas con una acción de exclusión, puede ser que aunque se cumpla la condición, el concepto que debe excluirse no está en la consulta original; sin embargo, se debe excluir cualquier contenido que contenga *Concepto1*, por los permisos que tienen los roles o porque es la preferencia del usuario.

excluir("Concepto₁, Concepto₂,..., Concepto_n")
Figura 46. Esquema de escritura de Exclusión

Siendo así, las reglas de equivalencia quedarían guardadas como se indica en la Figura 47.

```
<Reglas>
  <ReglaEquivalencia Condicion= "Consulta.contiene ("Restaurante")" Proceso = "adicionar ("Bogotá")" />
  <ReglaEquivalencia Condicion= "Consulta.contiene ("Ubicación")" Proceso = "reemplazar ("Ubicación, "localización")" />
</Reglas>
```

Figura 47. XML con reglas de equivalencia

Habiendo presentado la persistencia de las reglas de equivalencia, se define el esquema que se seguirá para almacenar las reglas de componente.

5.3.5.3.2. Persistencia de las reglas de componente

Cómo se dijo anteriormente, existen dos tipos de relaciones de componentes: *i*) las relaciones entre un componente y un valor y *ii*) las relaciones entre componentes. Dentro del archivo donde se almacenan las relaciones y reglas, las relaciones de componente tendrán una sección para cada una de estos tipos. Se denominarán relaciones de componente sencillas y complejas, respectivamente.

En las relaciones sencillas, se pondrá a la izquierda el valor con el que se desea comparar y a la derecha el componente con el cual se compara. La especificación de cómo se define el

componente y atributo a ser usado se muestra en la Figura 48 a, y el cómo se guardará la relación completa se especifica en la Figura 48 b.

$$\text{Perfil.Componente.atributo}_a = \text{Concepto1} = \text{Perfil.Componente.atributo}_b$$

Figura 48. Especificación de componente a comparar y de comparación

El segundo tipo de relaciones de componente que se puede definir es el que compara dos atributos de componentes distintos. La Figura 49 muestra el esquema de cómo establecer relaciones entre componentes.

$$\text{Perfil.Componente1.atributo} = \text{Perfil.Componente2.atributo}$$

Figura 49. Esquema de relación entre componentes

A partir de las relaciones se procede a establecer cómo se guardan las reglas. A continuación se define cómo escribir las reglas de cada de valor-componente y componente-componente.

Tal como se describió en las reglas de equivalencia, la condición y el proceso se guardan por separado; en este caso, la comparación se define entre el valor de un atributo y un valor ó entre el valor de dos atributos. El proceso se define de la misma manera que en las reglas de equivalencia. En la Figura 50 se muestra como se guardan las reglas de componente, en un archivo XML.

```
<Reglas>
  <ReglaComponente Condicion="Contexto_Actividad:localizacion='Universidad'" Proceso="adicionar:'Academico'" />
  <ReglaComponente Condicion="Contexto_EspacioTemporal:tiempo.hora='1400'" Proceso="adicionar:'Contexto_Actividad.localizacion'" />
</Reglas>
```

Figura 50. XML de reglas de componente

A continuación se procede a explicar la forma en que se realiza la priorización de las reglas.

5.3.6. Priorización de las reglas



Figura 51. La priorización de las reglas dentro del proceso

Como se observa en la Figura 51, a continuación se presentara la priorización de las reglas, para determinar cuáles serán usadas en el momento de enriquecer las consultas.

Las reglas de equivalencia se aplicarán siempre que se cumpla la precondition, mientras que las reglas de componente se aplicarán dependiendo de la priorización de sus componentes relacionados. Partiendo de esto, se define en principio la priorización a los componentes.

5.3.6.1. Priorización de componentes

Siguiendo la idea de cómo están representados los perfiles de sesión e histórico, se procede a plantear un método de priorización que aplicaría para la estructura de datos del árbol n-ario.

El método de priorización propuesto consiste en asignar una prioridad a cada rama del mismo. Para esto se deberá establecer un peso a cada una de las ramas, siguiendo el esquema propuesto por Panayiotou *et al.* [58], como se describió en el estado del arte (sección 3.1.1.5). A cada rama se le asignará un valor entre -1 y 10. Estos pesos simbolizan lo siguiente:

-1: indica que el usuario no desea tener en cuenta lo consignado en esa rama para hacer el enriquecimiento de consultas.

0: cuando el usuario establece que el peso de una preferencia es 0, indica que le da lo mismo si la información contenida en esa rama es incluida o no en el enriquecimiento.

1 a 10: Rango de valores para los componentes que se deben tener en cuenta para la adaptación de las consultas. Cada rama tendrá un peso establecido por el diseñador de la aplicación. Cabe aclarar que no es necesario que la suma de las prioridades de las ramas sea 10. Serán simplemente número entre uno y cien que el usuario asignará según su propio criterio.

Partiendo desde la raíz del árbol, la rama cuyo peso sea el más alto, y se realizará el mismo proceso hasta llegar a sus hojas. En la Figura 52 se muestra el árbol n-ario, y junto a cada componente se encuentra una letra, cada una de éstas será reemplazada por el peso asignado.

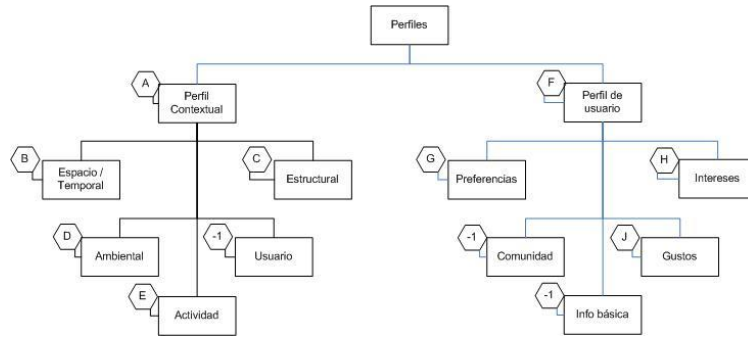


Figura 52. Asignación de pesos a los componentes

A partir del peso de los componentes se define la priorización de las reglas de componente.

5.3.6.2. Priorización de reglas de componente

Como se especifico anteriormente, las reglas de componente, parten de una relación entre dos de dichos componentes. Por lo tanto, se puede definir la prioridad de la reglas de componente, como el promedio de las prioridades de los componentes relacionados. Es decir, si una regla se define a partir de una relación entre el componente actividad (con prioridad 5) y el componente espacio temporal (con prioridad 3), se sugiere que las reglas resultantes tengan una prioridad de 4, debido a que este es el promedio de los componentes relacionado.

Habiendo descrito como se deben definir las prioridades, se procede a definir el perfil de sesión y el perfil histórico, de los cuales se adquiriran los datos necesarios para el enriquecimiento. A continuación se procede a explicar el perfil de sesión.

5.3.7. Perfil de sesión



Figura 53. El perfil de sesión dentro del proceso

Una sesión está definida como el lapso de tiempo, que inicia cuando un usuario ingresa a una aplicación que utiliza el sistema de enriquecimiento, hasta que se desconecta o cierra la aplicación.

A su vez, el perfil de sesión registra los cambios que se den mientras ésta dure, por medio de un registro de cambios. Este perfil está compuesto de un perfil contextual y un perfil de usuario.

En la Figura 54, se muestra la representación ontológica del perfil de sesión. Esta representación, hace extensible la arquitectura para los casos en que el diseñador de la aplicación considere hacer cambios sobre los perfiles propuestos en el presente trabajo.

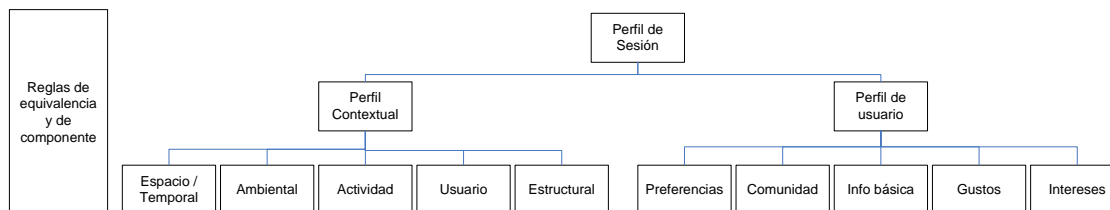


Figura 54. Representación del perfil de sesión

De acuerdo a la Figura 54, las reglas de equivalencia se representan sin un vínculo directo con el modelo de adaptación, debido a que definen sinonimia que no está relacionada de manera directa con algún componente particular. Las reglas de componente también se representan sin un vínculo directo con los componentes de los perfiles dado que las relaciones de las que se sirven están intrínsecas en su definición. El perfil contextual de la sesión, partirá de un esqueleto vacío como el definido aquí y se llenarán los datos mediante los métodos de adquisición ya descritos. Si mediante los métodos de adquisición no se logra completar todos los datos necesarios, entonces se podrá optar por ignorarlos o por preguntar al usuario. Lo que se haga depende de la priorización que se entregue al sistema de enriquecimiento de consultas. Si el usuario o el diseñador de la aplicación define que el componente espacio/temporal, por ejemplo, se usará para enriquecer las consultas pero no hay datos para el mismo, entonces se preguntará al usuario los datos necesarios.

El perfil de usuario cargará los datos que éste tenga registrados en el sistema. Estos datos se guardarán junto con el perfil histórico para que al iniciar una sesión se sepa exactamente dónde están y así se puedan cargar en el menor tiempo posible.

Es importante también, que al iniciar la sesión se carguen aquellas reglas de componente que se pueden aplicar según las características contextuales y de usuario. Esto con el fin de buscar más rápidamente las reglas que se puedan aplicar cuando el usuario ingresa una consulta. Sin embargo, es necesario que en el perfil de sesión se carguen todas las reglas de equivalencia ya que se aplicarán en el momento en que el usuario consulte y no antes, dado que no se sabe sobre qué conceptos se va a hacer la búsqueda de información.

5.3.8. Perfil histórico



Figura 55. Perfil histórico dentro del proceso

El perfil histórico se define como una recopilación de las consultas que un usuario ha realizado en sesiones anteriores. Al finalizar cada sesión, las consultas realizadas se guardarán para sugerir consultas relacionadas³⁴ al usuario. Es importante notar que se guardará solamente aquella información de los elementos de los perfiles que tengan prioridad (sea asignada por el usuario o por el diseñador de la aplicación).

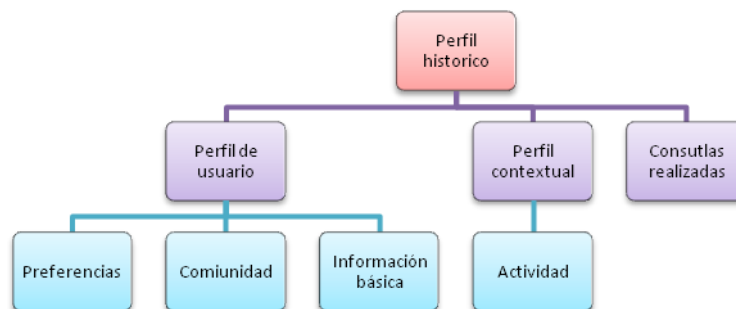


Figura 56. Representación del perfil histórico

La Figura 56 muestra la representación del perfil histórico en forma de árbol n-ario. Este árbol tendrá tres secciones:

La primera ramificación contendrá los datos perdurables del perfil de usuario. En él estarán almacenados los datos del usuario que no cambiarán con el tiempo, es decir, la información básica del usuario que le identifica ante el sistema así como las preferencias que se definen de acuerdo a las actividades y a las actividades relacionadas. También se almacenan los roles a los cuales pertenece el usuario, con el fin de usar esta información a la hora de aplicar las reglas que hacen control de contenidos.

³⁴ Cuando un usuario realiza una consulta Q_1 , y generalmente realiza una consulta Q_2 (dentro de la misma sesión), entonces la consulta Q_2 podría ser una consulta relacionada con Q_1 , dependiendo de la frecuencia (estadística)

La segunda rama, contendrá los datos perdurables del perfil contextual que se mantendrán a través del tiempo. Esto incluye las actividades que define el usuario, y cualquier otro elemento que se agregue a la definición del perfil y que tenga sentido (según el diseñador de la aplicación) para guardarlo a través del tiempo (e.g., una bitácora de los movimientos del usuario durante el curso del día).

La tercera rama del árbol, contendrá los datos de las consultas realizadas por el usuario (Q). Esto debido a que si se guardaran las consultas ya enriquecidas (Q'), las reglas con las que se hizo el proceso de enriquecimiento podrían ser obsoletas (puesto que las reglas pueden ser eliminadas, actualizadas o porque se podrían crear nuevas reglas) y la información consignada en el perfil histórico sería contradictoria a los requerimientos del usuario y/o aplicación en un momento específico.

Al terminar cada sesión, se almacenan las consultas que el usuario realizó, lo cual servirá para sugerir consultas relacionadas basadas en estadísticas. Por ejemplo, si se determina estadísticamente, que cuando el usuario consulta por “cámaras fotográficas”, también busca “centros históricos”, entonces, cuando el usuario va a consultar “centros históricos”, se le podría sugerir realizar una consulta sobre cámaras fotográficas.

5.3.9. Reescritura de consultas



Figura 57. La reescritura de consultas dentro del proceso

Como se describió en el estado del arte (Sección 3), una consulta *SQL*, cuenta con un *SELECT*, donde se especifican los valores que se desean buscar, un *FROM*, donde se especifica dónde se desea buscar la información y un *WHERE* donde se describen las restricciones de la consulta. Para modificar una consulta *SQL*, se deben tener en cuenta esas tres cláusulas. Asimismo, dentro del sistema de enriquecimiento de consultas se especificó la utilización de un módulo de enrutamiento que se encarga de seleccionar las fuentes de información que pueden responder la

consulta, debido a las diferencias que se presentan en los esquemas que manejan las fuentes de información.

A parte de esto, si bien la estructura de *SQL* para realizar consultas es estándar, la forma en que se organiza la información y los nombres de las entidades y atributos pueden ser diferentes de una base de datos a otra; por ejemplo, una misma entidad puede ser llamada en una base de datos “estudiante”, en otra “apl_estudiante” o “student”, y a su vez, un atributo podría ser el nombre completo del estudiante, mientras que en otra base de datos podría corresponder a dos atributos: nombre y apellido. Esta heterogeneidad en la organización de la información implica que al realizar la reescritura sea necesario conocer de antemano las fuentes de información y la forma en la que está se organizan los datos. De todo esto se concluye que no sea posible realizar reescrituras genéricas y se podrían generar pérdidas de información al utilizar las consultas históricas como insumo para la reescritura. Es decir, si se tiene en el histórico la búsqueda de “nombre” y esto se utiliza en una base de datos donde esté nombre y apellido, la información no sería completa y probablemente la consulta no arroje ningún tipo de resultado.

El sistema de enriquecimiento de consultas está pensado para poder realizar enriquecimiento y reescritura de consultas en el futuro. Las consultas en *SQL* poseen complejidad NP[27][49][19]. También es importante tener en cuenta lo complicadas que pueden llegar a ser las consultas en este tipo de lenguaje (*e.g.* un *join* entre tablas con miles de tuplas). La complejidad de las consultas y lo complicadas que estas pueden llegar a ser pueden causar tiempos de respuesta de varios minutos o incluso horas, durante los cuales algunos datos contextuales pueden haber cambiado y no ser útiles para la aplicación y para el usuario. Por lo tanto, una reescritura de tipo *SQL*, en la cual se especifica de dónde se debe adquirir la información y donde se debe conocer cómo se organiza y maneja la información, no es acorde al sistema propuesto en el presente trabajo ya que se estaría realizando también el trabajo del módulo de enrutamiento que sale del marco de este trabajo de grado. Es por esto, que el proceso de reescritura, busca modificar una cadena de caracteres, que se enviará al sistema de enrutamiento de un motor de búsqueda tradicional como es el caso de Google. Cabe aclarar, que si bien la reescritura del presente trabajo de grado, está definida como la modificación de una cadena de caracteres que se pasa a un motor de búsqueda, y que como éste es el encargado de enrutar dicha consulta, el tiempo de respuesta no será alto³⁵

Por otra parte, los motores de búsqueda utilizan cadenas de caracteres y conectores lógicos para las consultas [36]. Así, una reescritura en este tipo de aplicación se basa en la reconstrucción de

³⁵ Hace referencia a que se puedan presentar cambios en los datos contextuales mientras se realiza la recepción de resultados.

la cadena de caracteres, bien sea cambiando las palabras existentes por otras más acordes al contexto y a las necesidades del usuario o utilizando operadores lógicos a la consulta.

Para determinar qué motor de búsqueda podría ser utilizado como intermediario tanto para el enrutamiento como para la definición de los caracteres lógicos, se procedió a realizar una comparación de los motores de búsqueda más conocidos (ver Figura 58). A partir de estos valores se determinó que se utilizaría Google, debido a que es el motor de búsqueda más utilizado.

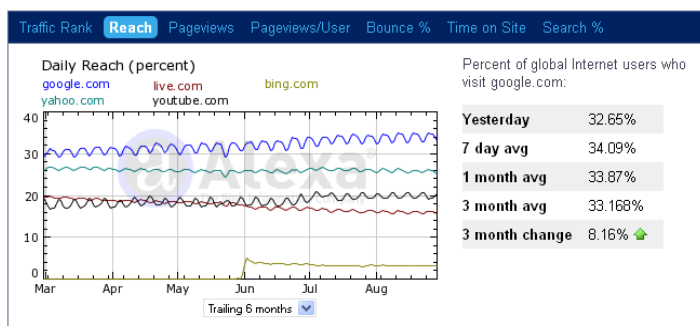


Figura 58. Porcentajes de ingreso a los motores de búsqueda [4]

A continuación se presentan los caracteres utilizados por *Google*TM y la descripción de los mismos.

Notación	Resultados
Palabra1 Palabra2	Palabra1 y Palabra2 (por defecto se considera la existencia de un AND entre dos palabras) ³⁶
Palabra1 OR Palabra2 Palabra1 Palabra2	Palabra1 ó Palabra2, o ambos
+ Palabra1	Buscar CON la palabra1 (es especialmente usado para palabras que el motor de búsquedas no toma en cuenta)
-Palabra1	Buscar resultados SIN la palabra1
~Palabra1	Buscar con la palabra1 o algún sinónimo.
Número i... Número n (e.g. 2001..2007) ³⁷	Buscar entre el rango dado
“Palabra1 * Palabra2”	Reemplazar * por una o más palabras, y buscar en el orden exacto
“Frase”	Buscar exactamente la frase entre comillas (“”)

Tabla 12. Adaptado de [36]

³⁶ Cuando en una búsqueda intervienen dos o más palabras y no se encuentran resultados que contengan todas las palabras, se hace una búsqueda considerando un OR entre dichas palabras.

³⁷ Para las reglas de equivalencia (sección 5.3.5) que utilicen rangos se utilizara esta convención.

Con el objetivo de realizar el proceso de reescritura de consultas, se deben utilizar las reglas previamente establecidas (ver sección 5.3.5), y así modificar los conceptos de la consulta ingresada por el usuario. En estas reglas se establece una precondition y si ésta se cumple, se realiza una acción con el concepto indicado.

Se pueden realizar tres acciones en dichas reglas, *i)* añadir un concepto por medio del operador “+”, *ii)* reemplazar un concepto de la consulta (*e.g.* si una aplicación considera pertinente establecer el concepto cerca como 50m, cerca se reemplazaría por el valor establecido dentro de una regla de equivalencia) y *iii)* la eliminación de un concepto por medio del operador “-“. A continuación se presenta una tabla con las posibles reglas y cómo se realizaría la reescritura.

Acción	Consulta Original	Regla	Consulta enriquecida
Adicionar	Concepto1	(Consulta.contiene(“Concepto1”)) → adicionar (“Concepto2”)	Concepto1 + Concepto2
Reemplazar	Concepto1	(Consulta.contiene(“Concepto1”)) → reemplazar(“Concepto1”, “Concepto2”)	Concepto2
	Concepto1 Concepto2	(Consulta.contiene(“Concepto1”)) → reemplazar(“Concepto2”, “Concepto3”)	Concepto1 Concepto3
Eliminar	Concepto1 Concepto2	((Consulta.contiene(“Concepto1”)) → eliminar (“Concepto1”)	Concepto2 - Concepto1

Tabla 13.Reescritura a partir de las reglas

5.3.10. Dinamismo del contexto



Figura 59. El dinamismo del contexto dentro del proceso

Para el presente trabajo de grado se define un registro de cambios, con el fin de reflejar las variaciones que pueden existir en el contexto desde el momento en que se realiza una consulta hasta el momento en el que se le entregan los resultados al usuario. Con este registro de cambios se contempla el dinamismo del contexto. También se definen las acciones a tomar

cuando los datos contextuales cambian y cómo fijar los márgenes de tolerancia para considerar significativos dichos cambios.

En la siguiente sección se presenta el proceso a seguir para llevar a cabo el registro de cambios.

5.3.10.1. Generalización del proceso

Para llevar a cabo el registro de cambios se deben ejecutar las siguientes actividades. Primero que todo se tiene que haber definido el perfil contextual que utilizará la aplicación; esto con el fin de definir los componentes considerados como dinámicos. Luego estos se priorizan y se define(n) la(s) acción(es) que se debe(n) llevar a cabo en el momento en el que se presenten estos cambios. Además de estos se deben definir los márgenes de tolerancia que tendrán los valores de los componentes para poder definir si los cambios son o no relevantes³⁸. Este proceso se resume en la Figura 60. Proceso de configuración del registro de cambios.

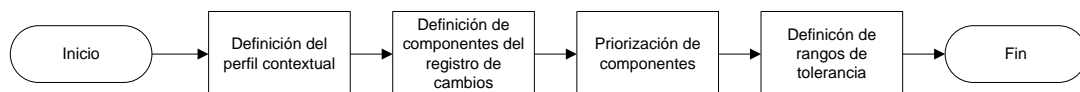


Figura 60. Proceso de configuración del registro de cambios

Luego de definir los componentes del registro de cambios y cómo se manejan, se explica cómo se utilizará el registro de cambios en la Figura 61. Proceso de uso del registro de cambios. Se comienza verificando la existencia de cambios en el contexto, para luego determinar si se debe modificar la consulta o si se entregan los resultados sin tener en cuenta el cambio contextual. Para esto se va a tener un número máximo de reescrituras que será definido por el diseñador de la aplicación, para evitar que se dé el caso en el cual siempre que se vayan a entregar los resultados el contexto varíe y sea necesario volver a realizar el proceso.

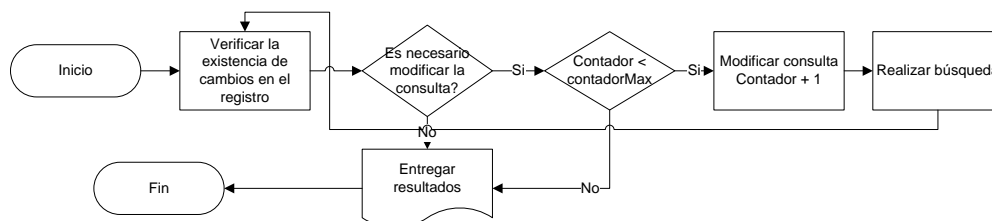


Figura 61. Proceso de uso del registro de cambios

A continuación se presentan los componentes que se incluyen en el registro de cambios, a partir de la instancia definida en el presente trabajo de grado.

³⁸ La relevancia en los cambios es definida por el diseñador de la aplicación.

5.3.10.2. Componentes del registro de cambio

En esta sección se muestra la representación ontológica del registro de cambios (ver Figura 62) y la descripción de cada elemento.



Figura 62. Ontología del registro de cambios

El registro de cambios está conformado por componentes de tipos estáticos o dinámicos.

- **Tiempo límite:** Es un valor numérico, medido en segundos. Cuando una consulta excede este tiempo de respuesta, los resultados podrían no ser pertinentes, bien sea por que se generan muchos cambios o por los requerimientos de la aplicación. Este tiempo es definido por el diseñador de la aplicación.
- **Número de iteraciones:** Define el número máximo de veces que se realizará el proceso de verificación de cambios en el registro y modificación de la consulta. Esto con el fin de evitar que el proceso pueda quedarse en un ciclo infinito. El número de iteraciones es definido por el diseñador de la aplicación.
- **Componentes estáticos:** Estos componentes son aquellos que no cambian su valor o que el cambio no es significativo para la aplicación.
- **Componentes Dinámicos:** Son los componentes cuyos posibles cambios deben ser tenidos en cuenta en la aplicación. Para cada componente se puede definir un margen de tolerancia.
 - **El margen de tolerancia:** Es un valor, con el cual se define los límites en los que un cambio es significativo. Este límite es usado en el caso de que un componente cambie su valor y se usa para determinar si es necesario cambiar los parámetros de la consulta es decir, reescribirla consulta con los nuevos valores contextuales, para poder entregarle al usuario resultados acordes a su nuevo contexto. Por ejemplo, se define un margen de tolerancia del componente tiempo, como un día (24 horas). Con lo cual se tendría que, si los resultados se entregan en un tiempo menor a 24 horas, **no** habría necesidad de hacer la reescritura del valor temporal del perfil contextual. En caso de que se excedan las 24 horas sería necesario realizar una reescritura.

Es decir, si se buscan las noticias del 3-08-2000 y se entrega un resultado del 4-08-2009, es probable que las necesidades del usuario hayan cambiado y sea necesario cambiar la consulta a noticias del 4-08-2009, pero si por el contrario los resultados se entregan con más de un día de diferencia será necesario considerar la posibilidad de cambiar el valor temporal de la consulta. Para más información remítase a la sección 5.3.10.5.

5.3.10.3. Conflictos

A continuación se describen los conflictos que se pueden generar al considerar ciertos cambios³⁹ en elementos de adaptación, ya sean contextuales o de perfil de usuario.

- **Conflicto de relaciones:** cuando un componente contextual cambia y se ha definido una relación con otros componentes (contextuales o del perfil de usuario), enseguida se define cómo actuar en caso de que el componente relacionado sea contextual o que pertenezca al perfil de usuario. A continuación se definen los conflictos de relaciones que se pueden presentar en cada perfil:
 - **Contextual:** Si existe una relación entre el componente que cambió (sin importar su prioridad) y otro(s) componente(s) contextual(es) y la prioridad del componente relacionado es 2, es necesario cambiar el valor del componente relacionado; si por el contrario el valor es 1, este cambio no afectará al componente relacionado.
 - **Perfil de Usuario:** Si existe una relación entre el componente que cambió y las preferencias del usuario, y a su vez dichas preferencias fueron incluidas en el enriquecimiento de la consulta, será necesario modificar dicho valor por el cambio producido con la relación. Por ejemplo, si el componente actividad está relacionado con el componente preferencias de actividad (ver secciones 2.3.1 y 2.4.1) y la actividad cambia de “trabajo” a “ocio”, es probable que la preferencia de actividad también cambie, por ejemplo, de buscar java lenguaje de programación, a buscar java café.

³⁹ para esto se tienen en cuenta los cambios que son pertinentes para la aplicación, a partir de los rangos de tolerancia y de la priorización de los componentes de adaptación implicados o relevantes para la aplicación.

5.3.10.4. Priorización de componentes

Para la priorización, el diseñador es el encargado de definir un peso entre 0 y 2 para cada componente. Cuando se especifica el peso de un componente como 0, este componente será estático. Si el valor es 1 indica que es importante considerarlo dinámico pero no será de mucha importancia. Si el valor es 2 indica que el cambio del componente afectará drásticamente los resultados, por ejemplo, una aplicación sensible a la localización tendría este componente con una prioridad 2.

Si se presenta un conflicto y el peso para los dos componentes es el mismo, se procede a preguntar al usuario, si desea aplicar alguno de los cambios o continuar con la consulta sin importar la variación presentada.

5.3.10.5. Margen de tolerancia

Habiendo definido los componentes que se incluyen en el registro de cambios y la priorización, se debe continuar como se especifican los márgenes de tolerancia. Como se puede notar en la Tabla 14, el margen de tolerancia se especifica respecto a un subcomponente, se define el sistema en que se medirá (por ejemplo para el componente temporal se podía definir, minutos, horas, días, entre otros), el valor del margen, es decir si se establece un margen de 24 horas, los cambios se consideran pertinentes si han pasado más de 24 horas para la recepción de resultados. Asimismo, se contara con un peso, que indica la prioridad del margen. Para esta prioridad se utilizaran los mismos rangos contemplados en la priorización de componentes.

En la tabla, también se presenta una ejemplificación la cual parte de un valor inicial, y el valor desde el cual un cambio sería pertinente, el cual es calculado a partir del margen de tolerancia definido.

Sub-Componente	Margen de tolerancia	Valor	Peso	Ejemplificación	
				Valor inicial	Valor desde el cual se tiene en cuenta un cambio
Temporal	Horas	24	0	2/12/2009, 10:00	2/12/2009, 10:00
Ubicación	Figura, Área, posición	Cuadrado, (500 km) ²	2	250	501
Temperatura	Grados	10	2	25	35
Nombre Actividad	Por defecto	Todos los cambios son tenidos en cuenta	1	Clase Algebra	Clase de programación

Tabla 14. Especificación de rangos de tolerancia

6. Caso de Estudio

Todas las contribuciones desarrolladas a partir de la investigación previa, fueron validadas por medio de un prototipo funcional. Con el fin de realizar dicha validación, se consideraron varias aplicaciones candidatas, para ser lo que en este trabajo denominamos caso de estudio. Las aplicaciones candidatas fueron: *i*) una aplicación de búsquedas en una biblioteca, *ii*) una agencia de viajes y *iii*) una aplicación de búsqueda de películas en un negocio de alquiler. De estas tres aplicaciones, se eligió la agencia de viajes, debido a que puede tener en cuenta una mayor información del contexto.

Por ejemplo, la aplicación de búsquedas en la biblioteca y en el negocio de alquiler de películas tendría en cuenta solamente los gustos e intereses del usuario, sin tener en cuenta otras de las características que rodean la interacción del usuario con el sistema, ya que al usuario le gustan las mismas películas sin importar el contexto en el que se desenvuelva. En el caso de la aplicación de una agencia de viajes, la inclusión de características contextuales en la búsqueda, como la actividad que se realizará en la fecha para la cual se busca el plan o la profesión del usuario, pueden influir en el tipo de plan que desean obtener, por ejemplo, si un usuario va a realizar un viaje con fines de negocio, los horarios de los vuelos deben ser en las primeras horas de la mañana o en los últimos vuelos programados en el día, para aprovechar mejor el día laboral, por medio de estos se pueden probar las reglas de equivalencia y de componente definidas para la aplicación.

La aplicación escogida, una versión Web de una agencia de viajes, se desarrolló como un prototipo funcional, por medio del cual se validan las contribuciones. En esta etapa de validación se hará uso de reglas de componente y de equivalencia definidas de acuerdo al dominio de la aplicación, y por lo tanto del modelo de adaptación propuesto, incluyendo los perfiles de usuario, sesión, históricos y de sesión. Con este fin se desarrollaron diversos casos de prueba, los cuales se explicaran más adelante, en esta sección.

La agencia de viajes tendría como fin la entrega de resultados personalizados sobre paquetes turísticos, pasajes de avión, costos y ubicación de hoteles; para esto se utilizó el sistema de enriquecimiento de consultas que toma las consultas que el usuario hace al sistema (e.g. al realizar una búsqueda de planes turísticos entre el 15 de febrero y el 2 de marzo), y usando reglas e información disponible enriquece la consulta (e.g. se buscarán sólo aquellos planes entre el 15 de febrero y el 2 de marzo que estén relacionados con “trabajo”, la actividad que estará realizando el usuario en este lapso de tiempo). Adicionalmente, se pueden realizar

búsquedas Web por medio de cadenas de caracteres, que luego enviar a Google™, para ser utilizado como módulo de enrutamiento de consultas. Para su parte, las búsquedas propias del sistema, como búsquedas de planes, se enviaron directamente a la base de datos que contiene la información. En resumen, cada uno de los servicios del prototipo buscó validar las contribuciones, al utilizar los componentes de la siguiente manera:

Servicio	Componentes usados
Búsqueda por Google™	Reglas de equivalencia
Promociones	Perfil histórico
Búsqueda por profesión	Perfil de usuario Reglas de componente

Tabla 15 Servicios y componentes usados

Todos los documentos asociados con el desarrollo del prototipo y del sistema de enriquecimiento de consultas (SRS, SDD, SAD), se podrán encontrar en el Anexo H.

6.1. Arquitectura del prototipo

El prototipo desarrollado cuenta con una arquitectura de tubos y filtros. Esta arquitectura está representada en la Figura 63.

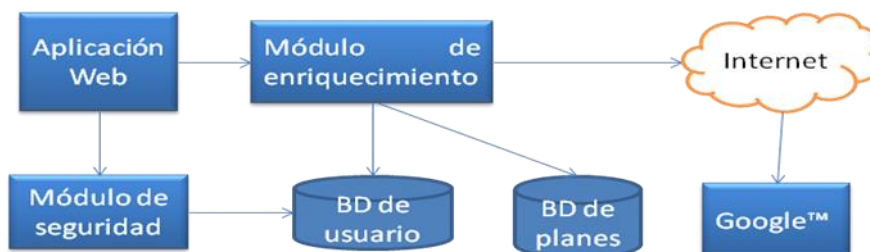


Figura 63. Arquitectura del prototipo VIVASUM

La arquitectura descrita en la Figura 63, cuenta con la fachada de VIVASUM, una aplicación web encargada de mostrar los contenidos a los usuarios y la cual se comunica mediante WCF (*Windows Communication Foundation*) a un módulo de seguridad, el cual se encarga de validar el ingreso de los usuarios al iniciar sesión, utilizando la base de datos de los usuarios en donde se guarda: el perfil de usuario y el perfil de contexto (toda aquella información persistente, es decir el perfil histórico).

La aplicación web se comunica con el módulo de enriquecimiento que toma las consultas originales (Q), y de acuerdo a la información almacenada en la base de datos de los usuarios y las reglas definidas para la aplicación (almacenadas en un archivo XML), se enriquece Q

dependiendo del servicio seleccionado (generando Q'), para luego enviar Q' a la base de datos de planes turísticos o a *GOOGLE™*, que retornará los resultados de la consulta enriquecida.

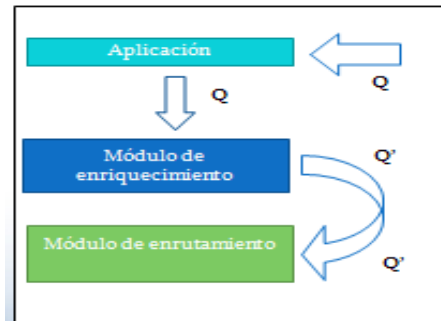


Figura 64 Sistema de enriquecimiento de consultas

El módulo de enriquecimiento mostrado en la Figura 63, se implementó en forma de *Framework*, es decir, que el sistema fue diseñado para ser utilizado por una aplicación diferente a *VIVASUM*, siempre y cuando se modifiquen las clases que contienen información sobre las consultas a la base de datos de la aplicación (Ver Figura 64), las clases a ser modificadas se encuentran especificadas en el manual de instalación, Anexo K. En la implementación que se hizo del módulo de enriquecimiento de consultas, existe una fachada que sirve para que las aplicaciones que lo usan se puedan comunicar con él y obtener los resultados, siendo así, los métodos ofrecidos en esta fachada pueden ser modificados para cumplir con los objetivos de la aplicación cliente. En el módulo de enriquecimiento desarrollado se hizo una implementación del perfil de usuario y de la agenda del usuario (parte del perfil contextual). Esta implementación se hizo para almacenar la información del usuario y su agenda en una base de datos. El diagrama de entidad relación se muestra en la Figura 65.

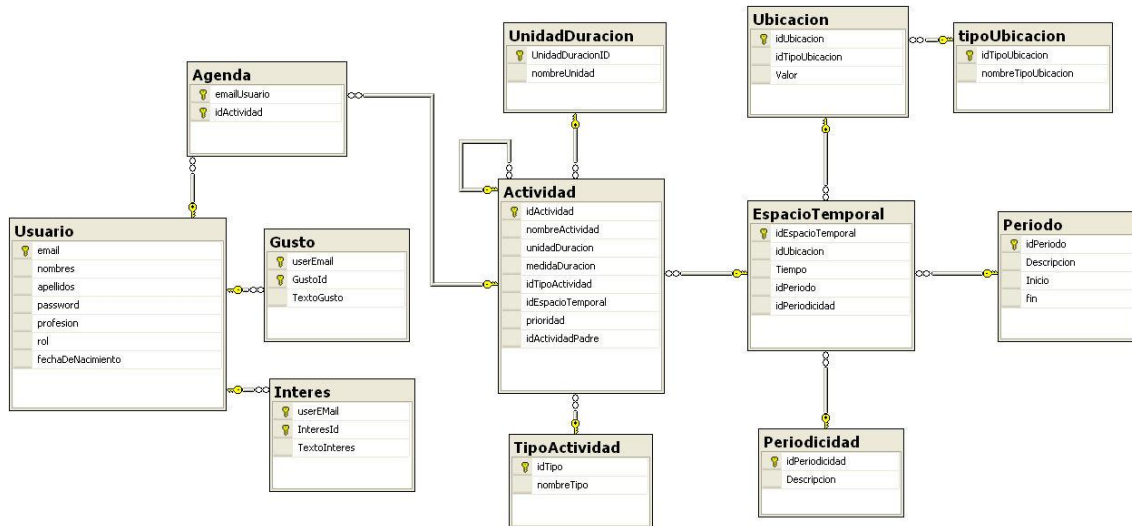


Figura 65. Diagrama entidad relación del módulo de enriquecimiento

Como la aplicación *VIVASUM* se usa para hacer búsquedas de planes turísticos, se implementó una base de datos que guarda la información de los mismos, cuyo diagrama Entidad Relación se muestra en la Figura 66.

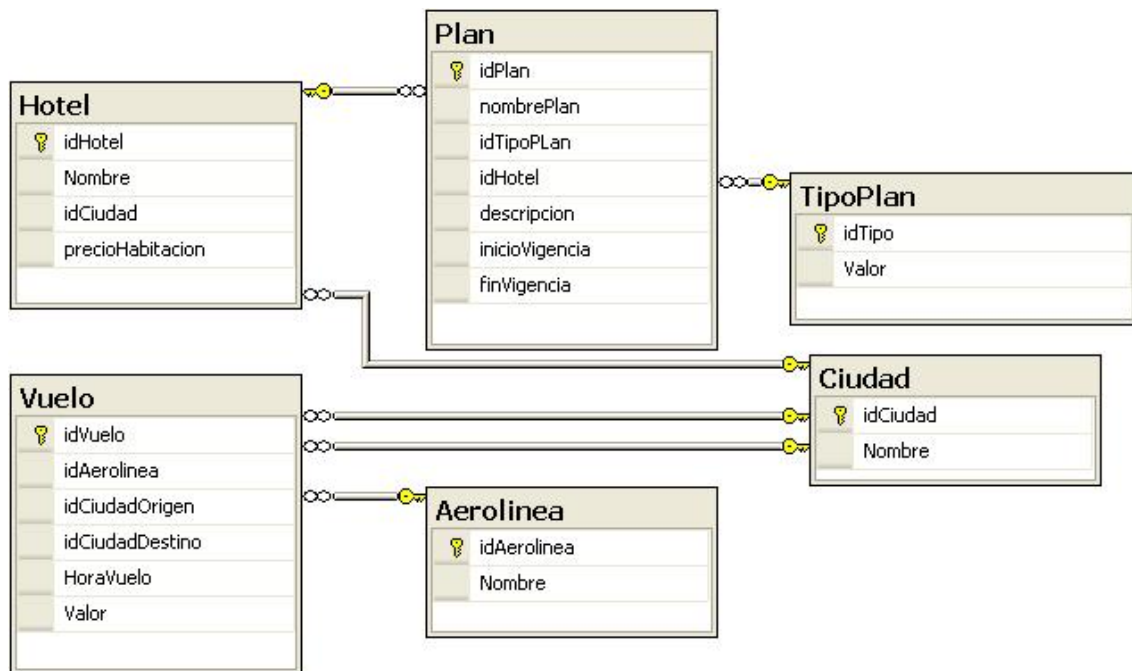


Figura 66. Diagrama Entidad Relación de la aplicación *VIVASUM*

Partiendo de la información consignada en el módulo de enriquecimiento de consultas, se realizan las búsquedas sobre la información almacenada en la base de datos de la aplicación (independientemente de la aplicación, aunque en este caso se habla de *VIVASUM* que es la aplicación desarrollada como prototipo).

Para que el módulo de enriquecimiento de consultas propuesto funcione, se hizo en código (C#.Net) una representación de las entidades de la Figura 65 (por medio del patrón *DAO*), esto con el fin de realizar un correcto acceso y manipulación de los datos contenidos en la base de datos, pensando especialmente en el uso en las reglas de componente. Estas reglas de componente y las de equivalencia, son evaluadas por partes en el módulo de enriquecimiento que las va desglosando. Primero se especifica si es una regla de componente o de equivalencia, luego se determina la condición y si esta se cumple o no. Si se cumple, se procede a ejecutar la lista de acciones presentes en la definición del proceso de la regla.

A su vez, se desarrolló una clase que se encarga de cargar en memoria las reglas de equivalencia y componente, especificadas para la aplicación particular, desde un archivo *XML*. Este archivo deberá ser editado por el diseñador de la aplicación para agregar manualmente las reglas nuevas siguiendo las pautas establecidas anteriormente en este documento y los ejemplos mostrados en el archivo “RuleInfo.xml”, que hace parte del diseño del prototipo. Cuando estas reglas son leídas y se encuentran en memoria, una clase llamada “Enriquecedor” es la que se encarga de hacer la reescritura de las consultas. Cuando la fachada del módulo de enriquecimiento de consultas recibe una consulta, Esta clase, “Enriquecedor”, hace la evaluación de las reglas contra los datos actuales e indica si éstas se cumplen o no, luego se realizan las acciones definidas en el proceso si la o las condiciones se cumplen.

6.2. Casos de uso

Para validar el funcionamiento del módulo de enriquecimiento de consultas se escogieron cuatro requerimientos. Estos casos de uso fueron: i) Búsqueda adaptada en Google™. Se hace con el fin de validar la funcionalidad de las reglas de equivalencia definidas para la aplicación. ii) Búsqueda de planes turísticos según la profesión del usuario. Se busca validar el uso de reglas de componente definidas respecto a elementos del perfil de usuario. iii) Búsqueda de planes turísticos según la actividad que el usuario vaya a estar realizando en la fecha para la cual hace la búsqueda de planes. Aquí se busca validar el uso de reglas de componente definidas respecto a elementos del perfil contextual y iv) sugerencias de planes turísticos de acuerdo a las consultas realizadas anteriormente. Se usa para validar la utilidad del perfil histórico en donde se han guardado las consultas anteriores realizadas por el usuario.

Cabe aclarar que a parte de los casos de uso anteriores se tuvieron en cuenta otros que ayudaron a prestar los servicios de la mejor manera (*e.g.* existe un caso de uso de ingreso en el sistema sin el cual no se podría saber quién es el usuario y no se podría cargar su información relacionada).

6.3. Pruebas funcionales

A partir del prototipo se realizaron algunas pruebas, cuyo documento detallado se encuentra en el Anexo I. A continuación se presentan los resultados más relevantes de dichas pruebas.

Los servicios brindados por la agencia de viajes (escenarios de prueba) son: *i*) búsquedas a través de *Google*TM, en donde se adaptada la consultas inicial del usuario, dependiendo de las reglas definidas por el diseñador de la aplicación; *ii*) búsqueda de planes según la actividad del usuario, en donde se muestran planes de ocio si el usuario se encuentra en vacaciones, planes de trabajo si el usuario tiene registrada en la agenda actividades laborarles y si no tienen ninguna actividad registrada se muestran todos los planes existentes; *iii*) búsqueda de planes por profesión del usuario, en donde se muestran los planes vacacionales más económicos si el usuario es un estudiante y todos los planes si no es un estudiante; y *iv*) el ofrecimiento de promociones basadas en el perfil histórico y el perfil de usuario, en este caso se ofrece un descuento de 5% para profesionales y 10% para estudiantes, en el plan más económico, que contemple la ciudad de origen y de destino, que el usuario consulta con mayor frecuencia.

6.4. Pruebas unitarias y de integración

Para el desarrollo de las pruebas, se utilizó Enterprise Architech, donde se realizaron diagrama de pruebas, dentro de los cuales se incluye la documentación de los casos de prueba realizados por cada caso de uso. Para las pruebas de integración, se corren los casos de prueba de varios servicios en la misma sesión.

Los casos de prueba contienen los siguientes cinco (5) componentes principales:

- Nombre, es el identificador único del caso de prueba. El formato para escribir el nombre es CU NumSecuencial- Tipo de prueba.
- Estado, indica si el caso de prueba ya fue corrido o no. Si ya fue corrido puede estar en “pass” que significa que el caso de prueba funciono correctamente o “fail” caso en el cual existe un bug y debe ser devuelto a al área de desarrollo. Si el caso de prueba no se corrió por alguna razón justificable, este se encontrara en estado Cancelado.
- Descripción, indica el objetivo del caso de prueba, los prerrequisitos por su parte, muestran las condiciones a cumplir para corre el caso de prueba y los pasos que se tienen para llevar a cabo el caso de prueba.
- Criterios de aceptación, si el caso de prueba esta correcto, la aplicación debe realizar lo especificado en dicho cambio

- Resultados, muestra el resultado obtenido al correr el caso, si se presenta algún error, se describe en este campo.

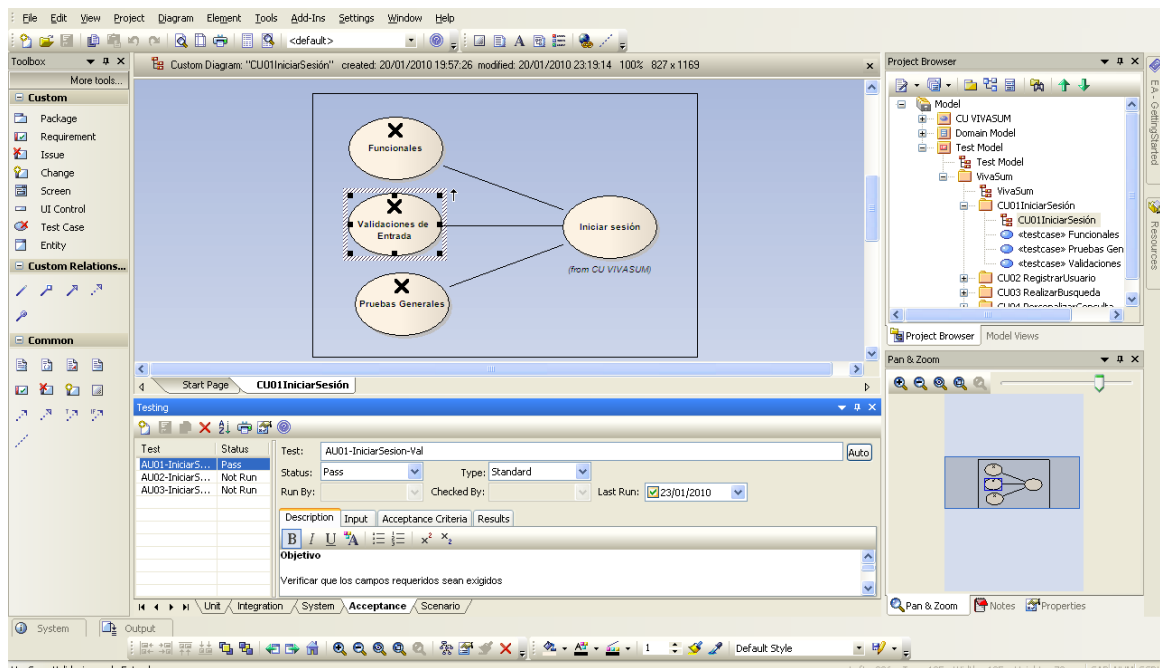


Figura 67 Pruebas por casos de uso

6.5. Tipos de prueba

Para validar el correcto funcionamiento y coherencia entre lo planteado en los documentos de casos de uso y requerimientos (tanto del sistema de enriquecimiento de consultas, como del prototipo funcional), se contó con las siguientes pruebas:

- **Funcionales:** Las pruebas funcionales buscan determinar si el flujo de la aplicación cumple con el camino de éxito o escenario feliz.
- **Validaciones de entrada:** Permiten verificar el formato de los campos, los nombres de los campos y los datos requeridos estén acordes a los especificado en los documentos de diseño.
- **Pruebas generales:** Buscan determinar si la imagen del producto es adecuada y la correcta navegación por las pantallas.

6.6. Resultados

Partiendo de los casos de prueba desarrollados en el Enterprise Architect, se obtuvieron los siguientes resultados:

En la Figura 68, se presenta el estado de los casos de prueba. Se conto con un total de 65 casos de prueba, los cuales fueron corridos o cancelados. De los 76 casos, un total de 67 fueron aprobados, bien sea luego de corregir los bugs o al mismo momento de la prueba. Por su parte se cancelaron 6 casos de prueba, por las razones anteriormente expuestas en este documento.

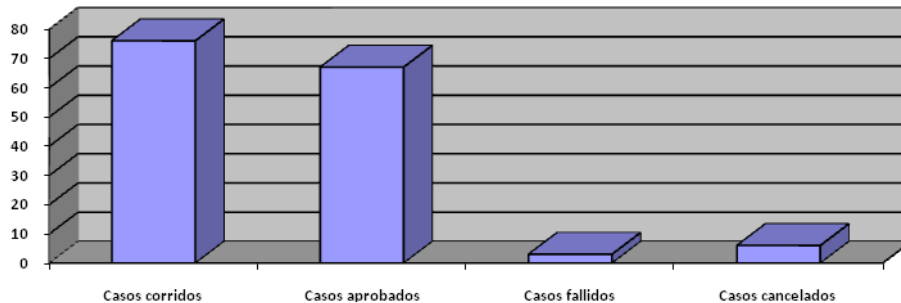


Figura 68 Estado de los casos de prueba

De todos los casos de prueba realizados, se conto con un 42% que validaban la funcionalidad del sistema, un 30% que verificaban las entradas y la forma en que el usuario ve el sistema y un 28% equivalen principalmente a pruebas respecto a la ortografía y navegación por la aplicación, es decir a las pruebas generales (ver Figura 69).

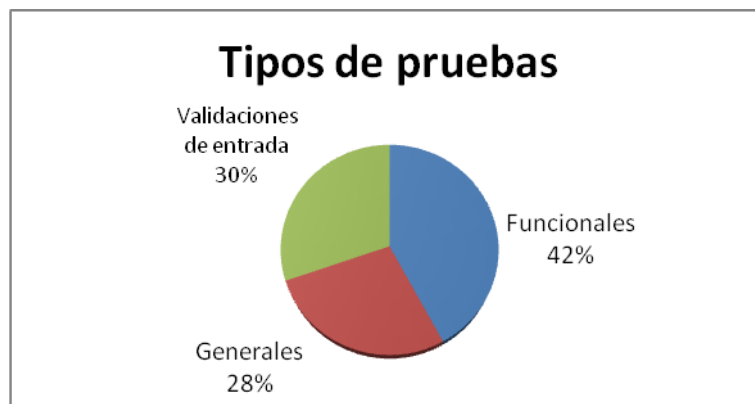


Figura 69 Estadística tipos de prueba

Respecto a la cantidad de errores (bugs) encontrados, la gran mayoría se identificaron al correr los casos de prueba, e inmediatamente fueron reportados para ser corregidos. Sin embargo, se encontraron otros errores mientras se corría la aplicación y se iban corrigiendo inmediatamente y por eso no se encuentran reportados en los casos de prueba. Algunos de estos errores son por ejemplo errores al realizar la conexión a la base de datos, o en la forma en que se leían los archivos XML, por la importancia y el carácter bloqueante de estos errores, se corregían inmediatamente (ver Figura 70). Cabe aclarar que todos los bus encontrados fueron corregidos en su totalidad.

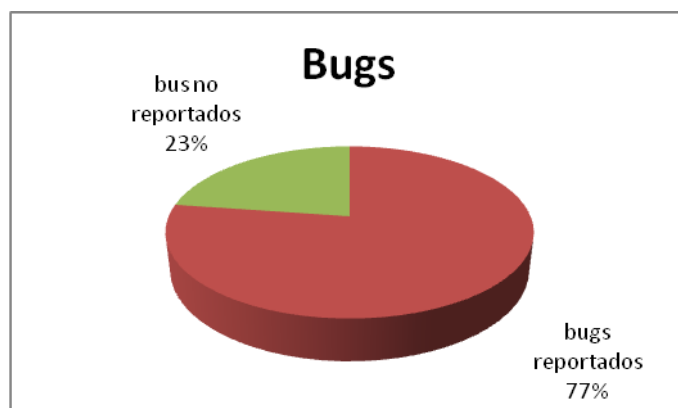


Figura 70 Cantidad de bugs

Al realizar las pruebas se identificaron algunos fallos como el no actualizar la actividad del usuario; a medida que se detectaron dichos problemas se fueron corrigiendo. Cabe destacar en este punto que cada uno de los cuatro servicios propuestos cumple con su objetivo, y sin embargo, se debe aclarar que no se probaron todas las funcionalidades descritas en el documento de casos de uso ni de especificación de requerimiento. Las características no probadas y la justificación se encuentran en la Tabla 16.

Característica no probada	Razón
Registro de cambios	Debido a que los tiempos de respuesta de las consultas no sobrepasan los 4 segundos, los cambios que se pueden presentar en los datos contextuales no son significativos para los fines del prototipo.
Márgenes de tolerancia	Hace parte del registro de cambios, al no ser probado el registro, tampoco se probó esta característica.
Cambios a los perfiles	Se considera que los elementos contenidos en los perfiles son suficientes para satisfacer las necesidades de la agencia de viajes.

Tabla 16. Características no probadas

6.7. Pruebas de usuario

Luego de finalizar las pruebas de funcionalidad, se probó el sistema con diferentes usuarios. Para esto, se les mostró el sistema *VIVASUM* y se describió el comportamiento general de los servicios de la aplicación, así como el objetivo al que se deseaba llegar. Luego de esto, se presentaron unas encuestas con las cuales se obtuvieron las siguientes recomendaciones:

- Realizar búsquedas sin personalización si el usuario así lo desea.
- Mejorar los colores de fondo
- Poder incluir más características, por ejemplo mostrar la fecha y hora actuales para “*crear un sentido de sintonía en el usuario*”⁴⁰
- Poder cambiar la fuente en la que se muestra el contenido (arial, times new roman)

⁴⁰ Comentario tomado textualmente de las encuestas realizadas en las pruebas de usabilidad.

- La adquisición de información debe ser con menos interacción (no pedir tantos datos)

7. Conclusiones y Trabajos futuros

Para finalizar el presente trabajo, se listan las conclusiones a las que se llegó a partir de la metodología realizada. A su vez, se identificaron ciertos trabajos futuros, con los que se puede ampliar el alcance del trabajo. En primera instancia, se presentarán las conclusiones, para luego continuar con los trabajos futuros.

7.1. Conclusiones

A partir de la investigación presentada en el estado del arte, se vio que la mayoría de los perfiles presentados (tanto de usuario como de contexto), están diseñados para una aplicación específica, lo que dificulta poderlos adaptar para su uso en otras aplicaciones. A su vez, se percibió que, para personalizar las consultas realizadas por un usuario, casi siempre se tiene en cuenta un perfil de usuario y un enriquecimiento semántico (*i.e.* un enriquecimiento de acuerdo al significado de las palabras incluidas en la consulta, buscando sinónimos o palabras con un significado similar), pero pocas veces se considera la necesidad de incluir las características contextuales (perfil contextual) o de enriquecer la consulta sintácticamente.

Con el fin de realizar una adaptación al contenido, se estableció un modelo de adaptación, que incluye el uso de un perfil de usuario, donde se especifiquen las características del usuario y un perfil de contexto, que se encargue de describir la interacción del usuario y el sistema. El fin de utilizar conjuntamente estos dos perfiles, es poder adaptar el contenido, no sólo respecto a los gustos del usuario, sino también a las necesidades que éste posee en un momento determinado (contexto), en este punto cabe aclarar que las necesidades de un usuario pueden variar dependiendo del contexto en el que se encuentre.

Al utilizar datos contextuales para el enriquecimiento de las consultas, se estableció la necesidad de contar con un registro de cambios, debido a que los datos contextuales pueden variar desde el momento en que se realiza la consulta, hasta que se entregan los resultados al usuario. Asimismo, estos cambios se pueden presentar constantemente, como es el caso del elemento temporal, que cambia su valor cada segundo; partiendo de esta idea, se estableció la necesidad de contar con márgenes de tolerancia, para así poder identificar cuando un cambio contextual podría influir en que los resultados entregados al usuario no satisfagan las necesidades del mismo.

A partir de la investigación realizada, se propuso un perfil de usuario y uno de contexto, representado por medio de ontologías; sin embargo, los perfiles propuestos están sujetos a

modificaciones, para que sean adaptados dependiendo de las necesidades de la aplicación que utilice el sistema de enriquecimiento. Para poder realizar dichos cambios, se establecieron ciertas tareas que el diseñador de la aplicación cliente debería considerar en el momento de modificar, eliminar y añadir nuevos componentes.

Debido a la complejidad de la reescritura de consultas en *SQL*, por la heterogeneidad de las fuentes de información, forma en que es tratada la información y la complejidad algorítmica inherente a formular una sentencia *SQL* bajo estas condiciones, se estableció una reescritura de consultas formadas por cadenas de caracteres, que después de enriquecidas se envían a un motor de búsqueda para encontrar sus resultados.

Para realizar el enriquecimiento de consultas, se introdujeron dos tipos de reglas que permiten la reescritura: *i*) las reglas de equivalencia que se encargaron de definir la reescritura semántica y sintáctica y *ii*) las reglas de componente, las cuales realizan el enriquecimiento a partir de los datos consignados en los perfiles. Luego de utilizar las reglas, esta se redirige a un módulo de enrutamiento.

A partir del trabajo realizado, se escribieron dos artículos. Uno basado en la propuesta de grado y publicado en el Congreso latinoamericano de estudios en informática (*CLEI*) 2008 y que lleva por título “Módulo de enriquecimiento de consultas basados en perfiles de usuario y de contexto”. El segundo artículo “Aspectos a considerar para adaptar el contenido y el despliegue de información”, desarrollado conjuntamente con un proyecto especial de la Pontificia Universidad Javeriana, donde se definió un modelo de adaptación, que incluye: un perfil de usuario, un perfil de contexto, un perfil de comunidad, un perfil de despliegue y un módulo de conexión inalámbrica, fue publicado en la revista Avances en Sistemas e Informática (*RASI*).

7.2. Trabajos futuros

Luego de realizar la investigación previa, las contribuciones y el prototipo utilizado para probar el sistema de enriquecimiento de consultas, se consideraron ciertas perspectivas que podrían mejorar la adaptación. A continuación se presentan los trabajos que podrían aumentar el alcance del trabajo:

- Existen otros perfiles que podrían contribuir en la adaptación al contenido. Por ejemplo, se podría considerar un perfil grupal que contenga información más específica acerca de la comunidad donde se encuentra el usuario. También se podría contar con un módulo

de conexión inalámbrica, el cual sería útil para identificar por ejemplo el ancho de banda y establecer el tamaño máximo de los resultados que se entreguen al usuario.

- Este trabajo se centró en la adaptación al contenido, pero como se había descrito en el estado del arte, también existe una adaptación al despliegue. Este tipo de adaptación se podría incluir en el sistema, al utilizar por ejemplo un perfil que indique las características del dispositivo de acceso, las preferencias de despliegue del usuario y las características de la conexión que se está utilizando, con el fin de presentar los resultados que sean compatibles con el dispositivo de acceso y con las preferencias del usuario. Por ejemplo, un usuario puede definir que prefiere recibir archivos en formato “.avi”, y desde que la conexión inalámbrica y las características del dispositivo lo permitan, se pueden entregar resultado en dicho formato.
- Para el manejo del perfil histórico, se propone el uso de algoritmos de minería de datos. El uso de este tipo de herramientas permitiría inferir información de posibles comportamientos del usuario y algunos datos no conocidos, lo cual permitiría sugerir búsquedas relacionadas, o inferir los valores de ciertos elementos de los perfiles, en lugar de preguntarlos directamente al usuario.
- Debido al alcance del trabajo, se realizó la reescritura de consultas a partir de una cadena de caracteres. El sistema de enriquecimiento de consultas está diseñado para que pueda funcionar a partir de las reglas, con consultas *SQL*; es por esto que uno de los trabajos futuros es realizar la reescritura con consultas tipo *SQL* de forma generalizada, es decir, una reescritura *SQL* que pueda ser utilizada en cualquier sistema de información que lo maneje.

8. Referencia

- [1] Abbas, K., Verdier, C., Flori. A.: Exploiting Profile Modeling for Web-Based Information System, WISE 2007 Workshops, pp. 313--324. Springer Verlag Berlin Heidelberg (2007)
- [2] Abiteboul, S., Benjelloun, O., Milo, T.: Active XML and Active Query Answer. In: Flexible Query Answering System, LCNS, vol 3055, pp. 17--24. Springer Verlag, Berlin Heidelberg (2004)
- [3] Agostini, A., Moro, G.: Identification of Communities of Peers by Trust and Reputation. In: Bussler, C., Fensel, D. (eds.) AIMS 2004, LNCS, vol. 3192, pp. 85--95. Springer Verlag Berlin Heidelberg (2004)
- [4] Alexa the Web Information Company, <http://www.alexa.com/siteinfo/google.com>, última consulta Septiembre 2009
- [5] Arezki, R., Poncelet, P., Dray, G.: Pearson, D, Web Information Retrieval Based on User Profile. In: Adaptive Hypermedia and Adaptive Web-Based System, LNCS, vol 3137, pp. 17--27. Springer Verlag Berlin Heidelberg (2004)
- [6] Arocena, G., Mendelzon, A., Mihaila, G.: Applications of a web query Language, <http://www.cs.toronto.edu/~websql/www-conf/wsql/PAPER267.html>, última consulta noviembre de 2009
- [7] Aslam, A., Khan, S., Latif, K.: Semantic Based Query Rewriting in Heterogeneous Source. In: 4th International Conference on Emerging Technologies, pp. 292—297, Islamabad (2008)
- [8] Bai, J. , Nie, J., Bouchard, H., Cao, G.: Using Query Contexts in Information Retrieval. In: The 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp.12--22. Amsterdam, The Netherlands (2007)
- [9] Barkhuus, L., Dey, A.: Is Context-Aware Computing Taking Control Away from the User? Three Levels of Interactivity Examined. In: The 5th International Conference on Ubiquitous Computing (ubicomp), pp. 149—156. Seattle, Washington (2003)
- [10] Berhe, G., Brunie, L., Pierson J.: Modeling Service-Based Multimedia Content Adaptation in Pervasive Computing. In: 1st conference on computing frontiers, pp. 60--69. New York, USA (2004)
- [11] Berkovsky, S.: Ubiquitous User Modeling in Recommender Systems. In: User Modeling 2005, LNCS, vol 3538, pp. 496-498. Springer Verlag Berlin Heidelberg (2005)

- [12] Berti, L., Demoiseaux, J., Muriasco, E.: Combining the Power of Query Languages and Search Engines for On-line Document and Information Retrieval: The qiri@D Environment, In: Principles of Digital Document Processing, LNCS, vol 1481, pp. 112—127. Springer Verlag Berlin Heidelberg (1998)
- [13] Bielikovi, M., Adaptive Presentation of Evolving Information Using XML, In: IEEE International Conference on Advanced Learning Technologies, pp. 193—196. Washington, DC, USA (2001)
- [14] Bouzeghoub, M., Kostadinov, D. Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils. Actes de CORIA 2005, pp. 201-218. Grenoble, France (2005)
- [15] Brunkhorst, I., Dhraief, H., Kemper, A., Nedjl W., Wiesner, C.: Distributed Queries and Query Optimization in Schema-Based P2P Systems. In: Aberer, K., Kalogeraki, V., Koubarakis, M. (eds.): Proceedings of the 1st International Workshop on Databases, Information Systems, and Peer-to-Peer Computing (DBISP2P) (Berlin, Germany September 7-8, 2003), LNCS, vol. 2944, pp. 184-199. Springer Verlag Berlin Heidelberg (2003)
- [16] Brusilovsky, P.: Adaptive Hypermedia: From Intelligent Tutoring Systems to Web Based Education. In: Gauthier, G., Frasson, C., vanlehn, K. (eds.) ITS 2000, LNCS, vol. 1839, pp. 1--7, Springer Verlag Berlin Heidelberg (2000).
- [17] Bucur, O., Beaune, P., Boissier, O.: Steps Towards making Contextualized Decisions: how to do what you can, with what you have, where you are. In: Roth-Berghofer, T., Schulz, S., Leake, D.B. (eds.): Proceedings of the 2nd International Workshop on Modeling and Retrieval of Context (MRC 2005) (Edinburgh, UK, July 31 - August 1, 2005) Revised Selected Papers, LNCS, vol. 3946, pp. 62--85. Springer Verlag Berlin Heidelberg (2006),
- [18] Burak, A., Sharon, T.: Usage Patterns of friendzone - Mobile Location-Based Community Services. In: 3rd International Conference on Mobile and Ubiquitous Multimedia, vol. 83, pp. 93–100. New York, USA (2004)
- [19] Calvanese, D., De Giacomo, G., Lenzerini, M., Vardi, M.: What is Query Rewriting?. In: Cooperative Information Agents IV - The Future of Information Agents in Cyberspace, vol. 1860, pp. 51--59. Springer Verlag Berlin Heidelberg (2000)
- [20] Carillo Ramos, A., Gensel, J., Villanova-Oliver, M., Martin, H., PUMAS: a Framework based on Ubiquitous Agents for Accesing Web Information System through Mobile Devices. Proceedings of 20th Annual ACM Symposium on Applied Computing (SAC2005) (Santa Fe, USA, March 13-17, 2005), ACM Press, pp. 1003-1008, New York, NY (2005).

- [21] Carmichael, J., Kay, J., Kummerfeld, B.: Consistent Modelling of Users, Devices and Sensors in a Ubiquitous Computing Environment. Vol 15, pp. 197-234. Springer Netherlands (2005)
- [22] Chen, X., Chen, Y., Rao, F.: An Efficient Spatial Publish/Subscribe System for Intelligent Location-Based Services. In: 2nd International Workshop on Distributed Event-Based Systems (DEBS '03); Disponible en: http://www.eecg.toronto.edu/debs03/papers/chen_etal_debs03.pdf.
- [23] Cristancho, J., Romero, A., MAICO: Modelo de adaptación de la información en comunidades, Proyecto especial Pontificia Universidad Javeriana (2008)
- [24] Cui, H., Wen, J., Nie, J., Ma, W.: Query Expansion by Mining User LOGS. In: IEEE Transactions on Knowledge and Data Engineering, vol 15, pp. 829 -- 839. IEEE (2003)
- [25] Dan, Z.: Data Mining Application in the Banking Industry in China (1998-2007). In: International Conference on Information Management, Innovation Management and Industrial Engineering, vol 1, pp. 240, 243. IEEE (2008)
- [26] Dey, A.N., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: Gellersen, H.W. (eds.) HUC'99, LNCS, vol. 1707, pp. 304--307, Springer Verlag Berlin Heidelberg (1999)
- [27] Finance, B., Gardarin G.: A Rule-Based Query Rewriter in an Extensible DMBS. In: Seventh International conference on Data Engineering, pp. 248—256. Kobe, Japon (1991)
- [28] Gandon, F., Sadeh, N.: Semantic Web Technologies to Reconcile Privacy and Context Awareness. In: Journal of Web Semantics. Vol. 1, no. 3. (2004).
- [29] Gelbukh, A.: Lazy Query Enrichment: A Method for Indexing Large Specialized Document Bases with Morphology and Concept Hierarchy. In: Database and experts System Application, LNCS, vol 1873, pp. 526--535. Springer Verlag Berlin Heidelberg (2000)
- [30] Gregory, D., Abowd, D., Dey, A., Orr, R., Brotherton, J.: Context-Awareness in Wearable and Ubiquitous Computing. In: 1st IEEE International Symposium on Wearable Computers (ISWC), pp. 179 --180. IEEE Computer society, Washington (1997)
- [31] Grine, H., Delot, T., Lecinte, S.: Adaptive Query Processing in Mobile Environment. In: 3rd international workshop on Middleware for pervasive and ad-hoc computing, vol 115, pp.1--8. New York, USA (2005)

- [32] Heckmann, D., Krueger, Antonio.: A User Modeling Markup Language (userml) for Ubiquitous Computing. In: User modeling 2003, LNCS, Vol 2702, pp. 148--152. Springer Verlag Berlin Heidelberg (2003)
- [33] Henze, N., Nejdil, W., Knowledge Modeling for Open Adaptive Hypermedia. In: debra, P., Brusilovsky P., Conejo, R. (eds.): AH 2002, LNCS, vol. 2347, pp. 174-183, Springer Verlag Berlin Heidelberg (2002)
- [34] Higuera, M., Aragón, F., Carrillo, A.: Módulo de enriquecimiento de consultas basado en perfiles de usuario y contexto. In: XXXIV Congreso Latinoamericano de Estudios en Informática (CLEI 2008), pp. 559-568. Argentina (2008)
- [35] Hristova, N., P. O'Hare, G.M.: Ad-me: Intelligent Context-Sensitive Advertising within a Mobile Tourist Guide. In: 12th Irish AI and Cognitive Science Conference. Ireland (2001)
- [36] Interactive online Google tutorial and Reference, www.googleguide.com , última consulta Septiembre 2009
- [37] Jameson, A.: Modeling Both the Context and the User, Personal Ubiquitous Computing. In: Personal and Ubiquitous Computing , vol. 5, pp 29--33. Springer Verlag Berlin Heidelberg (2001)
- [38] Jin, H., Ning, X., Chen, H., Yin, Z.: Efficient Query Routing for Information Retrieval in Semantic Overlays. In: 2006 ACM symposium on Applied computing, pp. 1669--1673. New York, USA (2006)
- [39] Kassab, R., Lamirel, J.C., Nauer, E. Une nouvelle approche pour la modélisation du profil de l'utilisateur dans les systèmes de filtrage d'information basés sur le contenu : le modèle de filtre détecteur de nouveauté In : Actes de CORIA 2005, pp. 185-200. Grenoble, France (2005)
- [40] Kim, B., Kim, K.: Query Processing to Efficient Search in Ubiquitous Computing. In: Computational Science -ICCS 2007, vol 4490, pp. 849--852. Springer Verlag Berlin Heidelberg (2007)
- [41] Kim, H., Bu, K., Kim, J., Lee, S.: Exploiting Question Concepts for Query Expansion. Computational Linguistics and Intelligent Text Processing, LNCS, vol 3106, pp. 624--627. Springer Verlag Berlin Heidelberg (2005)
- [42] Kirsch-Pinheiro, M., Gensel, J., & Martin, H. Representing Context for an Adaptive Awareness Mechanism. Proceedings of the 10th International Workshop on Groupware (CRIWG 2004), LNCS, Vol. 3198, pp. 339-348. Springer Verlag Berlin Heidelberg (2004)
- [43] Kokkinidis, G., Christophides, V.: Semantic Query Routing and Processing in P2P Database System: The ICS-FORTH sqpair Middleware. In: Lindner, W., Masiti,

- M., Türker, C., Tzitzikas, Y., Vakali, A. (eds.) EDBT 2004, LNCS, vol. 3268, pp. 486--495, Springer Verlag Berlin Heidelberg (2004)
- [44] Koloniari, G., Pitoura, E., Content-Based Routing of Path Queries in Pair-to-Pair Systems. In: Lindner, W., Masiti, M., Türker, C., Tzitzikas, Y., Vakali, A. (eds.) EDBT 2004, LNCS, vol. 3268, pp. 29--47, Springer Verlag Berlin Heidelberg (2004)
- [45] Lech, T., Wienhofen, L.: Ambieagents: A Scalable Infrastructure for Mobile and Context-Aware Information Services. In: Aarts, H., Westra, J. (eds.): Proceedings of the 4th International Conference on Autonomous Agent and Multi-Agent Systems (AAMAS 2005) (Utrecht, Netherlands, July 25-29, 2005), pp. 625--631. ACM Press. New York, NY (2005)
- [46] Li, J., Guo, M., Tian, S.: A New Approach to Query Expansion. In: IV International Conference on Machine Learning and Cybernetics, IEEE, vol 4, pp. 2302, 2306. IEEE (2005)
- [47] Li, L.: A Query Expansion Method Based on Semantic Element. In: VIII International Conference on Software Engineering, IEEE, vol 1, pp. 587-590. IEEE (2007)
- [48] Li, Y., Yao, Y. Y.: User Profile Model: A View from Artificial Intelligence. In: Rough Sets and Current Trends in Computing, LNCS, vol 2475, pp. 83 – 87. Springer Verlag Berlin Heidelberg (2002)
- [49] Liu, J., Huang, F., Ye, D., Huang, T.: Efficient Consistent Query Answering Based on Attribute Deletions. In: International Symposium on Computer Science and its Applications, 2008, pp. 222--227. IEEE (2008)
- [50] Ma, L., Chen, Q., Cai, L.: An Improved Framework for Online Adaptive Information Filtering. In: Advances in Web-Age Information Management, LCNS, vol 2762, pp. 409-420. Springer Verlag Berlin Heidelberg (2003)
- [51] Medina-Medina, N., Molina-Ortiz, F., García-Cabrera, L., Parets-Llorca, J.: Personalized Guided Routes in an Adaptive Evolutionary Hypermedia System. In: 9th International Workshop on Computer Aided Systems Theory on Computer Aided Systems Theory (EUROCAST 2003) (Las Palmas de Gran Canaria, Spain, February 24-28, 2003), LNCS, vol. 2809, pp. 196-207. Springer Verlag Berlin Heidelberg (2003)
- [52] Michalski, R., Kaufman, K., Pietrzykowski, J., 'Snie'zy'nski, B., Wojtusiak, J.: Learning Symbolic User Models for Intrusion Detection: A Method and Initial Results. Advances in Soft Computing. Vol 35, pp. 273--285. Springer Verlag Berlin Heidelberg (2006)

- [53] Moon, N., Singh, R., Designing Interaction Paradigms for Web-Information Search and Retrieval, In: Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, pp. 815-822. Hong Kong (2006)
- [54] Murray, T., Piemonte, J., Khan, S., Shen, T., Condit, C.: Evaluating the Need for Intelligence in an Adaptive Hypermedia System. In: Gauthier, G., Frasson, C., Frasson, C.(eds) ITS 2000, LNCS, vol 1839, pp. 373-382, Springer Verlag Berlin Heidelberg (2000)
- [55] Neff, F., Kehoe, A., Pitt, I.: User Modeling to Support the Development of an Auditory Help System. In: Text, Speech and Dialogue, LNCS, vol 4629, pp.390--397, Springer Verlag Berlin Heidelberg (2007)
- [56] Norton, P.; Introducción a la computación. Mc GrawHill (2000)
- [57] Orozco, A., Cárdenas, J., Flórez, L., Carrillo, A.: PAMPU: Modelo de Preferencia de Actividades para la Definición de un Perfil. In: IV Congreso Colombiano de Computación, Bucaramanga, Colombia (2009)
- [58] Panayiotou, C., Samaras, G.: Mobile User Personalization with Dynamic Profiles: Time and Activity. In: OTM Workshops 2006, LNCS, vol 4278, pp. 1295--1304. Springer Verlag Berlin Heidelberg (2006)
- [59] Park, J., Barber, S.: Finding Information Sources for Model Sharing in Open Multi-Agent System. In: ubiagents04. Disponible en: <http://www.ift.ulaval.ca/~mellouli/ubiagents04/>
- [60] Pawlak, Z., Data Mining- a Rough Set Perspective. In: Third pacific-Asia Conference on methodologies for knowledge, discovery and Data mining, vol 1574, pp. 3-12, 1999. Springer Verlag Berlin Heidelberg (1999)
- [61] Pérez, J., Araujo L.: Query Expansion with an Automatically Generated Thesaurus. In: Intelligent Data Engineering and Automated Learning – IDEAL 2006, LNCS, vol 4224, pp. 771--778, Springer Verlag Berlin Heidelberg (2006)
- [62] Petrakis, Y., Koloniari, G., Pitoura E.: On using histograms as routing indexes in peer-to-peer systems. In: 2nd International Workshop on Databases, Information Systems, and Peer-to-Peer Computing (DBISP2P 2004), volume 2992, LNCS, pp. 16--30, Toronto, Springer (2004).
- [63] Portal Bogotá, www.bogota.gov.co, última consulta septiembre 2009
- [64] Ramirez, J., Gómez, R.: Generación de mediadores pedagógicos para facilitar el proceso de enseñanza – aprendizaje en la realización de las consultas a las Bases de Datos en la Pontificia Universidad Javeriana Bogotá. Trabajo de grado (2009).

- [65] Ruiz, F., Hilera, J.: Using Ontologies in Software Engineering and Technology. In: *Ontologies in Software Engineering and Software Technology*, pp. 49-102. Springer Verlag Berlin Heidelberg (2006).
- [66] Shapira, B., Hanani, U., Raveh A., Shoval, P.: Information Filtering: A New Two-Phase Model Using Stereotypic User Profiling. In: *Jornal of Intelligent Information Systems*, vol 8, pp. 155--165. Springer Netherlands (1997)
- [67] Siberski W., Thaden, U.: A simulation Framework for Schema-Based Query Routing in P2P Network. In: Lindner, W., Masiti, M., Türker, C., Tzitzikas, Y., Vakali, A. (eds.): *EDBT 2004, LNCS*, vol. 3268, pp. 436--445, Springer, Heidelberg (2004)
- [68] Skinner, B., *Contingencias del reforzamiento: un análisis*. México (1979)
- [69] Storey, V., Sugumaran, V., Burton, A.: The Role of User Profiles in Context-Aware Query Processing for the Semantic Web. In: *Natural Language Processing and Information System, LNCS*, vol 3136, pp.45--65. Springer Verlag Berlin Heidelberg (2004)
- [70] Sunikka, A. Bragge, J. : What, Who and Where: Insights into Personalization. In: *41st Annual Hawaii International Conference on System Sciences*, pp. 283--283. Waikoloa, IEEE (2008),
- [71] Tamine, L., Bahsoun, W. Définition d'un profil multidimensionnel de l'utilisateur. *Actes de CORIA 2006 (Lyon, France, 15-17 mars, 2006)*, pp. 225--236. Lyon, France (2006)
- [72] Tempich, C., Staab, S., REMINDIN': Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphors. In: *13th International Conference on World Wide Web*, pp. 640--649. ACM, New York USA (2004)
- [73] Tempich, C., Staab, S.: Semantic Query Routing in Unstructured Networks Using Social Metaphors. In: *Semantic Web and Peer-to-Peer*, pp. 107--123. Springer Berlin Heidelberg (2006)
- [74] Villanova, M. Adaptabilité dans les systèmes d'information sur le web : Modélisation et mise en oeuvre de l'accès progressif. Tesis Doctoral (En Francés), INPG, France, 2002.
- [75] Virgilio, R., Torlone, R.: Management of Heterogeneous Profiles in Context-Aware Adaptive Information System. In: *Proceedings of On the Move to Meaningful Internet Systems Workshops*, October 31-November 4, 2005, Agia Napa, Cyprus. (2005) pp. 132--141

- [76] Wang, G., Hu, J., Zhang, Q., Liu, X., Zhou, J.: Granular Computing Based Data Mining in the Views of the Rough Set and fuzzy set. In: IEEE International Conference on Granular Computing, 2008, pp. 67-67. IEEE (2008)
- [77] Wang, M., Xu, H., Hao, G., Zhou, X., Wang, W., Zhang, Q., Shi, B.: Picturebook: A Text-and-Image Summary System for Web Search Result. In: Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pp. 1612--1615. IEEE (2008)
- [78] Wang, Y., Aroyo, L. M., Stash, N., Rutledge, L.: Interactive User Modeling for Personalized Access to Museum Collections: The Rijksmuseum Case Study. In: Semantic Web and Peer to Peer, LNCS, Vol 4511, pp. 385--389. Springer Berlin (2007)
- [79] Yang, D., Xu, L., Cai, W., Zhou, S., Zhou, A., Efficient Query Routing for XML Documents Retrieval in Unstructured Peer-to-Peer Networks. In: Advanced Web Technologies and Applications, vol 3007, pp. 217--223. Springer Berlin (2004)
- [80] Yang, X., Wang, B., Wang, G., Yu, G.: A Query Rewriting System for Enhancing the Queriability of Form-based Interface. In: Digital Libraries: International collaboration and Cross-Fertilization, LNCS, vol 3334, pp. 462--472. Springer Berlin (2004)
- [81] Ya-Wen H., Moon, N. Singh, R.: Designing Interaction Paradigms for Web-Information Search and Retrieval. In: Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, 2006, pp. 815--822 (2006)
- [82] Yudelson, M., Gavrilova, T., and Brusilovsky, P. Towards user modeling meta-ontology. In: L. Ardissono, P. Brna and A. Mitrovic (eds.) Proceedings of 10th International User Modeling Conference (Edinburgh, UK, July 24-29, 2005), pp. 448-452. Berlin Springer (2005)
- [83] Zayanil, C., Péninou, A., Canut, M., Sedes, F.: An adaptation approach: query enrichment by user profile. In: Advanced Internet Based system and application, vol 4879, pp. 351 -- 361. Springer Berlin (2009)
- [84] Zhu, Z., Xu, J., Ren, X., Tian, Y.: Query Expansion Based on a Personalized Web Search, In: Third International Conference on Semantics, Knowledge and Grid, pp. 128-133 (2007)

9. Anexos

A continuación se presenta una lista con los anexos que se pueden descargar de la página Web:

- Anexo A. Glosario
- Anexo B. Tabla de convenciones usadas en los diagramas de flujo.
- Anexo C. Trabajos relacionados con el perfil de usuario, perfil de contexto, la reescritura de consultas y el enriquecimiento de consultas
- Anexo D. MAIPU
- Anexo E. MAICO
- Anexo F. MOCA
- Anexo G. Representación ontológica de los perfiles en OWL
- Anexo H. Documentos del prototipo y del sistema de enriquecimiento de consultas (Casos de uso, SRS, SDD, SAD, manuales)
- Anexo I. Documento de pruebas
- Anexo J. Referencias en BibText
- Anexo K . Manual de instalación