



Universidad de Granada

TESIS DOCTORAL

**Sistemas de Bases de Datos Difusas
Sensibles al Contexto**

José Tomás Cadenas Lucero

Directores:

Dr. Nicolás Marín Ruíz

Dra. María Amparo Vila Miranda

Programa Oficial de Doctorado en Tecnologías de la Información y la
Comunicación

Departamento de Ciencias de la Computación e Inteligencia Artificial
ETS de Ingenierías Informática y de Telecomunicación

Granada, Mayo de 2015

Editor: Universidad de Granada.Tesis Doctorales
Autor: José Tomás Cadenas Lucero
ISBN: 978-84-9125-180-4
URI: <http://hdl.handle.net/10481/40442>

El doctorando D. José Tomás Cadenas Lucero y los directores de la tesis Dr. Nicolás Marín Ruiz y Dra. María Amparo Vila Miranda garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, 22 de Mayo de 2015

Directores de la Tesis

Doctorando

Fdo.:

Fdo.:

Dedicatoria

A Dios,

A mi amante esposa y mis queridos hijos,

A mi madre y la memoria de mi padre

Agradecimientos

*Un agradecimiento muy especial para mis directores de tesis **M^a Amparo Vila Miranda y Nicolás Marín Ruiz**, sin su apoyo incondicional, profesionalidad, dedicación y esmero no habría sido posible la realización de este trabajo. Más allá de su labor de directores de tesis también han sido amigos y me ayudaron a afrontar con éxito esta ardua labor.*

*A **Leonid** por haber hecho posible mi relación con el Grupo de Investigación de Bases de Datos y Sistemas Inteligentes de la Universidad de Granada, dirigido por **M^a Amparo**; además de todo su apoyo profesional, como amigo y guía espiritual. No tengo palabras para agradecerles todo lo que han hecho por mí **Amparo, Nicolás y Leo**.*

*Doy gracias a **Jesús** por haberlos cruzado en el camino de mi vida, traerme a esta maravillosa ciudad, Granada, y poder hacer amigos de muchas partes del mundo.*

*A la **Fundación Carolina** y la **Universidad Simón Bolívar** por haberme apoyado durante mis estudios de doctorado con la beca otorgada; sin ella habría sido imposible para mí desarrollar el doctorado fuera de mi país de origen.*

A los profesores del Departamento de Ciencias de la Computación e Inteligencia Artificial, y a los del Máster en Soft Computing y Sistemas Inteligentes de la Universidad de Granada, así como ponentes en los múltiples seminarios que pude asistir y que contribuyeron con mi formación.

*Un agradecimiento especial al Prof. **Carlos Torrealba** por su gran ayuda en la corrección del inglés en escritos técnicos. A profesores y estudiantes de la Universidad Simón Bolívar que me apoyaron durante la realización de mi trabajo: **Rosseline, Soraya, Edna, Marlene, Rubén, Andreína, Verónica, Jhosbert, Daniela, Arturo, Hancel**, entre otros.*

*A mi madre **Mireya**, mi esposa **Macringer**, mis hijos **Tomás Daniel, Mariamni** y **José Miguel** por su apoyo, cariño y comprensión en momentos difíciles. A mis hermanas **Liliana** y **Evelyn**, mis hermanos **Luis Enrique, José Daniel, Miguel Ángel**. Mi familia política **Omaira, Gerardo** (padre e hijo), **Zoraya, Ibel** y **Nicolasa**.*

*A **Andrés Domenech** por su compañía y amistad durante mis estancias en Granada. A sus familiares, en especial su madre **Rosa María**, por haber tenido confianza en mí el primer día que nos conocimos a las puertas de la Facultad de ETS de Ingenierías Informática. A su abuela **Rosa María** por su cariño. A **Paco Rivas** por su amistad y apoyo cuando lo necesité.*

A todos los que de una forma u otra me ayudaron a lograr esta maravillosa meta y no los mencioné... ¡Mil bendiciones!

Índice General

1. Introducción	1
1.1. Planteamiento del Problema	3
1.2. Objetivo General	7
1.3. Objetivos Específicos	8
1.4. Metodología.....	8
1.5. Estructura del Trabajo	9
2. Estudio Preliminar	11
2.1. El Contexto.....	11
2.1.1. El Contexto en Computación.....	13
2.1.2. Ideas generales acerca del Contexto en Sistemas de Información y Bases de Datos	16
2.2. Aplicaciones Sensibles al Contexto	18
2.2.1. Consideraciones Generales	18
2.2.2. Comportamiento de las Aplicaciones Sensibles al Contexto.....	21
2.3. Sistemas de Información Sensibles al Contexto.....	22
2.3.1. Ideas Generales	23
2.3.2. Procesamiento de Consultas Sensibles al Contexto.....	24
2.4. Modelado del Contexto	25
2.5. Discusión	27
3. Bases de Datos Difusas	31
3.1.1. Conceptos Básicos	38
3.1.2. La Lógica Difusa en Representación de la Información.....	43
3.1.3. Bases de Datos Relacionales Difusas	44
3.1.4. Bases de Datos Orientadas a Objeto Difusas.....	47
3.1.5. El modelo de datos de Objetos Difusos de referencia	54
4. Sistema de BD Sensible al Contexto	65
4.1. Esquema de Interacción.....	66
4.2. El Contexto.....	67
4.2.1. Contexto relativo al usuario.....	68

4.2.2.	Contexto relativo al ambiente	70
4.2.3.	Otros.....	71
4.3.	Sensibilidad al contexto de la salida del sistema.....	71
4.3.1.	Semántica.....	72
4.3.2.	Resultado	72
4.4.	Arquitectura del Sistema de Bases de Datos Sensible al Contexto.....	74
4.4.1.	Integración de la sensibilidad al Contexto en Sistemas Informáticos.....	74
4.4.2.	Integración del Contexto en un SGBD	78
4.4.3.	Nuestra propuesta de arquitectura.....	80
4.5.	Aportes	81
5.	Atributos Lingüísticos con Semántica Adaptable al Contexto	83
5.1.	El problema de los atributos sensibles al contexto en Bases de Datos Difusas	84
5.1.1.	Dominios Difusos	85
5.1.2.	El problema de la Semántica	87
5.2.	Modelo Formal Propuesto	88
5.3.	Implementación transparente en un SGBDOR	92
5.3.1.	Gestor de Apoyo a la Sensibilización al Contexto	94
5.3.2.	Módulos CTP.....	95
5.4.	Conclusiones del Capítulo.....	107
6.	Aplicación Real de un SBDDSC	108
6.1.	Descripción de la Aplicación Real en SQL.....	108
6.1.1.	Dominio de la Aplicación.....	108
6.1.2.	El Examen Físico Articular.....	110
6.1.3.	Implementación de la Aplicación real en SQL.....	115
6.2.	Diseño de la Aplicación Web	121
6.2.1.	Plataforma de Desarrollo	121
6.2.2.	Lenguajes de Programación.....	122
6.2.3.	Base de Datos.....	122
6.2.4.	Interfaces de la Aplicación	125
6.2.5.	Consultas.....	131
6.2.6.	Sesión de Administrador.....	136

6.3. Utilización de la Arquitectura Propuesta en la Aplicación EFA-HOI .	138
7. Conclusiones y Trabajos Futuros	141
7.1. Conclusiones	141
7.2. Trabajos Futuros	145
8. Bibliografía	147
Apéndices	
A. Formularios Examen Físico Articular	159
B. Diccionario de Datos del Examen Físico Articular del Hospital Ortopédico Infantil	167

Índice de Figuras

1. El contexto y el ciclo computacional, tomado de Lieberman y Selker.....	19
2. Función de Pertenencia de una persona de estatura “alta”.....	39
3. Interpretación de número razonable de hijos en una familia	39
4. Interpretación de “alrededor de 50 años de edad”	40
5. FPs de valores lingüísticos “maduro”, “joven” y “viejo”	41
6. Valores lingüísticos triangulares usuales, adaptado de Zadeh	42
7. Dominios difusos para valores de atributos	56
8. Etiquetas lingüísticas para el atributo estatura	57
9. FP de la etiqueta lingüística “maduro”	58
10. FP de la etiqueta lingüística “viejo”	59
11. FP de la etiqueta lingüística “joven”	59
12. FP triangular para la función “alrededor de Y ”	61
13. Esquema de interacción de un usuario con un Sistema de Bases de Datos Sensible al Contexto.....	66
14. Modelo de contexto en un esquema de interacción Usuario con SBDSC	67
15. Arquitectura de un Sistema Sensible al Contexto	75
16. Modelo Contextual por Capas.....	76
17. Modelo Contextual por Capas propuesto	77
18. Conectando el Contexto y las Bases De Datos	79
19. Arquitectura para el desarrollo de un Sistema de Bases de Datos Sensible al Contexto	80
20. Jerarquía de Dominios Difusos	85
21. Modelo Sensibilización al Contexto en Sistemas de Información.....	88
22. Modelo de Dependencias Contextuales	92
23. Arquitectura para Sistemas de Bases de Datos Difusas Sensibles al Contexto	93
24. Gestor de Apoyo a la Sensibilización al Contexto.....	95
25. Esquema de Interacción dos usuarios en una <i>BDDSC</i>	96
26. Algoritmo de Inserción implementado por el CAS.....	97
27. Conjunto de etiquetas para Flexión de la Cadera. Contexto: <i>usuario1</i> , cualquier <i>tarea</i>	102

28. Conjunto de etiquetas para Flexión de la Cadera. Contexto: <i>usuario2</i> , <i>postoperatorio</i>	102
29. Ilustración del cálculo de <i>Fuzzy Equal (FEQ)</i>	104
30. Transformación Lingüística para el <i>Usuario1</i>	106
31. Transformación Lingüística para el <i>Usuario2</i>	107
32. Esquema General de la Aplicación Real	109
33. Caso de Uso Examen Físico Articular, Hospital Ortopédico Infantil	111
34. Aplicación Real de BD Difusa Sensible al Contexto.....	113
35. Diagrama de Clases para la implementación de la Aplicación Real.....	114
36. Muestra de valores almacenados en el Catalogo Contextual	116
37. Diagrama de Clases del EFA	123
38. Diagrama de Clases Dominios Difusos.....	123
39. Diagrama de Componentes de la Aplicación	124
40. Página de Inicio de Sesión para la Aplicación Web	125
41. Página de Registro.....	126
42. Inicio Interfaz Móvil	127
43. Datos Personales del médico.....	128
44. Etiquetas lingüísticas para el Peso	128
45. Matrices de semejanza para Tono Flexor Dorsal.....	129
46. Ingreso de un nuevo EFA.....	130
47. Ingreso Datos EFA interfaz móvil	130
48. Ingreso de datos Imperfectos	131
49. Datos de un Paciente	132
50. Resultado de la Consulta de Pacientes <i>Delgados</i> con <i>Marcha Normal</i>	133
51. Resultado de una Comparación entre Pacientes.....	133
52. Resultado de la Consulta “Pacientes con Peso <i>Normal</i> ” para el “usuario1”. 134	
53. Resultado de la Consulta “Pacientes con Peso <i>Normal</i> ” para el “usuario2”. 134	
54. Consulta Difusa del Examen Físico Articular con semántica de Peso y Talla del “usuario 1”	135
55. Consulta Difusa del Examen Físico Articular con semántica de Peso y Talla del “usuario 2”	135
56. Interfaz para ingreso de dominios difusos	137
57. Interfaz para ingreso de dimensiones contextuales.....	137

58. Interfaz para ingreso de trapezoide por defecto	138
59. Utilización de la Arquitectura Propuesta en la prueba de concepto.....	139

Índice de Definiciones

1. Conjunto difuso y función de pertenencia	38
2. Variable lingüística	41
3. Contexto de interacción entre un usuario y un SBDSC	68
4. Dimensión Contextual.....	89
5. Dominio no dependiente del Contexto.....	89
6. Dominio dependiente del Contexto.....	90
7. Conjunto de dependencias de un dominio	90
8. Atributo dependiente del contexto	90
9. Conjunto de dependencias de un atributo	90
10. Conjunto de dependencias de un dominio	91
11. Atributo dependientes del contexto.....	91
12. Expresión Lingüística.....	104

Índice de Ejemplos

1. Personas de estatura “alta”	38
2. Conjuntos difusos con un dominio discreto no ordenado	38
3. Conjuntos difusos con un dominio discreto ordenado	39
4. Conjuntos difusos con un dominio continuo.....	39
5. Variables lingüísticas y valores lingüísticos	41
6. Dominios atómicos con representación semántica asociada definidos sobre un referencial finito	56
7. Dominios atómicos con representación semántica asociada definidos sobre un referencial continuo	57
8. Dominios atómicos con representación semántica asociada definidos sobre un referencial continuo definido por funciones.....	58
9. Calidad de un producto	61
10. Idiomas de un investigador	63
11. Dominio Difuso Atómico Categórico	86
12. Dominio Difuso Atómico Discreto	86
13. Dominio Difuso Atómico Continuo.....	87
14. Dominio Difuso Conjuntivo.....	87
15. Utilización de la sentencia CREATE en una BDDSC	98
16. Dependencias contextuales para el dominio difuso Flexión de la Cadera....	99
17. Utilización de una sentencia INSERT por <i>Usuario1</i> proporcionando una etiqueta lingüística.....	101
18. Utilización de una sentencia INSERT por <i>Usuario2</i> proporcionando una etiqueta lingüística.....	101
19. Conjunto de etiquetas de <i>Usuario1</i> para el dominio difuso Flexión de la Cadera.....	101
20. Conjunto de etiquetas de <i>Usuario2</i> para el dominio difuso Flexión de la Cadera.....	102
21. Utilización del método LSHOW por parte del <i>Usuario1</i>	105
22. Utilización del método LSHOW por parte del <i>Usuario2</i>	106

Resumen

El avance de las Tecnologías de la Información y Comunicación, además de sistemas móviles y sensores presentes en el mundo real, producen una gran cantidad de datos que deben ser gestionados adecuadamente para extraer información que sea de utilidad para las personas que interactúan con los sistemas automatizados. Es así que el desarrollo de mecanismos que faciliten la Interacción Persona Ordenador es un tema de gran interés en el desarrollo de software y en particular en Sistemas de gestión de bases de datos. Además, un problema de investigación en esta área que sigue teniendo relevancia, el cual no puede ser abordado adecuadamente utilizando sistemas tradicionales, es el paradigma de coincidencia exacta. Es por ello que ha surgido la necesidad de desarrollar sistemas inteligentes, flexibles y personalizados al usuario. Una manera de contribuir a desarrollar este tipo de sistemas es mediante la utilización del contexto del usuario en el momento de la interacción con el sistema. Asimismo, se reconoce la importancia de modelar procesos de naturaleza humana, tal como la comunicación entre seres humanos, donde normalmente la información es imperfecta. La teoría de conjuntos difusos proporciona mecanismos para modelar y representar datos imperfectos es por ello el desarrollo de las Bases de Datos Difusas. En esta investigación estudiamos el contexto en aplicaciones de bases de datos y proponemos una arquitectura para el desarrollo de Sistemas de Bases de Datos Sensibles al Contexto que sirva de marco para el desarrollo de sistemas inteligentes, flexibles y personalizados al usuario. Definimos el modelo de contexto enfocándonos en el esquema de interacción usuario y sistema de gestión de base de datos. También se aporta un modelo formal para la representación y manipulación de atributos lingüísticos con semántica adaptable al contexto y su implementación transparente en un SGBDOR como Oracle con capacidades de gestión de datos difusos. El uso de los modelos propuestos permite representar datos de la vida real y manipularlos de manera amigable y sensible al contexto por parte de los múltiples usuarios de un sistema de información. A través de una aplicación real del área médica, utilizando modelos de datos orientados a objetos, dominios difusos y etiquetas lingüísticas, se demuestra la factibilidad de la utilización de la arquitectura y los modelos propuestos. Así, se presenta una implementación de una prueba de concepto en el área médica utilizando el Sistema Gestor de Base de Datos Objeto-Relacional (SGBDOR) Oracle, que puede ser usada por conocedores del estándar SQL:99 o a través de una aplicación web con interfaz móvil amigable para usuarios inexpertos.

Palabras claves: Aplicaciones sensibles al contexto, Bases de Datos Difusas, Sistemas inteligentes de base de datos, Sistemas Personalizados, SGBDOR

Capítulo 1

Introducción

Los sistemas de bases de datos tradicionales han funcionado bien por muchos años, convirtiéndose en una parte esencial de considerables desarrollos informáticos e implantándose en forma extensiva. Además, son de gran importancia para la toma de decisiones de una organización, proporcionando herramientas para analizar grandes cantidades de datos. Los sistemas de bases de datos dan soporte a gran cantidad de transacciones, asimismo poseen alta fiabilidad y disponibilidad.

Estamos viviendo en una era donde las organizaciones, en general, recolectan, mantienen y usan grandes cantidades de información acerca de sus clientes, proveedores y operaciones, mientras que con la ayuda del uso de dispositivos móviles, la web cada día es más amigable a todo tipo de usuario. Paradigmas emergentes como el Internet de las cosas, web de sensores, inteligencia ambiental, entornos inteligentes y las redes sociales contribuyen a complicar este panorama en cuanto a la cantidad de datos que se producen. También estos enfoques enfatizan la tendencia de realizar y mantener una conexión más fuerte entre las aplicaciones informáticas y el mundo real.

El gran volumen de datos proporcionado por la utilización de las tecnologías, si no se filtra en forma adecuada, agobia a los usuarios, perdiendo la oportunidad de utilizar el valioso conocimiento que puede ser extraído de estos datos. Además de las técnicas proporcionadas por diversas investigaciones (síntesis, comprensión y análisis de grandes volúmenes de datos), una forma de ayudar a resolver este problema es mediante la *personalización* de los sistemas dado que, de esta forma, se proporcionará al usuario información reducida tomando en cuenta las preferencias personales del mismo y/o su situación. En este sentido, el desarrollo de aplicaciones sensibles al contexto en Sistemas de Información, contribuye a mejorar la calidad del acceso a la información por parte del usuario mediante la personalización de aplicaciones.

En esta memoria presentaremos una propuesta de un modelo para el desarrollo de aplicaciones sensibles al contexto en Sistemas de Información, específicamente en el área de bases de datos, caracterizada por ser lo menos intrusiva y más transparente posible a los usuarios, logrado mediante la utilización de estándares reconocidos en dicha área; una de los propósitos es tratar de reproducir en el mundo virtual lo que sucede en el proceso de comunicación entre las personas.

Para ello, además tendremos en cuenta la imperfección presente en los datos proporcionados por el mundo real, especialmente los que son de naturaleza humana. Se plantea pues la incorporación de la borrosidad (*fuzziness*) en la representación de los datos, permitiendo, entre otras cosas, almacenar valores de variables lingüísticas y flexibilizar las condiciones de las consultas. Muchos investigadores han aplicado con éxito la teoría de conjuntos difusos de Zadeh [160] para modelar la imperfección de los datos en los sistemas gestores de bases de datos, acercándose de esta forma a una representación más fiel del lenguaje natural utilizado por el ser humano, dando lugar a lo que se conoce más ampliamente como Bases de datos Difusas [54]. Es por ello que se consideró de interés asumir esta teoría para la propuesta del modelo objeto de estudio.

Los sistemas de bases de datos tradicionales sufren del problema de *rigidez* debido a que responden a las consultas en forma precisa y determinista (paradigma de consulta de “coincidencia exacta” [77]). En este caso, una consulta obtiene respuestas (si las hay) que cumplen con todas las condiciones establecidas en dicha consulta. Además, el Sistema Gestor de Bases de Datos (SGBD) debe garantizar que se obtendrá *siempre* la misma respuesta a una consulta.

Este problema del paradigma de consulta de coincidencia-exacta acarrea dos inconvenientes principales: 1) La respuesta vacía, que puede abordarse mediante la relajación de algunas de las restricciones rígidas en la consulta, es decir, considerando restricciones flexibles, relajando necesidades del usuario o sustituyéndolas por otras que se ajusten más adecuadamente a sus preferencias; 2) Demasiadas respuestas, que puede ser afrontado mediante el fortalecimiento de la consulta con condiciones adicionales expresando las necesidades del usuario para limitar los resultados, personalizándolos.

En el presente trabajo se estudió la utilización de etiquetas lingüísticas en las bases de datos bajo el marco de la Teoría Computacional de Percepciones de Zadeh [168]. De esta forma se introduce un elemento enriquecedor del valor semántico en el proceso de las consultas, privilegiando la utilidad para el usuario de los resultados mediante la personalización basada en preferencias. En un caso de ejemplo, un término (tal como peso “normal”) puede tener un significado diferente dependiendo del contexto; de esta forma, la formulación de una misma consulta donde se utilice dicho término puede producir resultados diferentes, logrando *sensibilizar* los resultados de las consultas.

En las próximas secciones abordamos el problema de la personalización de sistemas de información, específicamente en el área de bases de datos a través de la incorporación del contexto; además, a través de la representación de datos imperfectos, incluyéndolo en el modelo de sensibilización al contexto, logramos dar un enfoque novedoso para el desarrollo de sistemas inteligentes, flexibles y personalizados al usuario.

1.1. Planteamiento del Problema

En la actualidad el software es considerado un factor crítico de éxito en la sociedad de la información y el conocimiento. Las relaciones entre las personas, empresas y gobiernos, han cambiado por el uso de la tecnología; un ejemplo claro de ello es la interconexión de las empresas con los clientes y sus asociados, utilizando herramientas de gestión del conocimiento para ser más eficientes; a la par se observa que los gobiernos emprenden acciones para mejorar su presencia a través de la web y prestar servicios cada vez más eficientes a los ciudadanos; mientras que los usuarios emplean las herramientas para sus relaciones interpersonales a través de las redes sociales, la web 2.0 ha conformado una nueva sociedad virtual, donde el eje fundamental es la información.

Las Tecnologías de la Información y las Comunicaciones (TIC) se encuentran incrustadas en las estructuras sociales y culturales en múltiples formas; además, se ha extendido su utilización más allá de áreas empresariales, destacando en la educación (*e-learning*), servicios médicos (*Telehomecare*) y en el hogar (*domótica*), donde diversos dispositivos computacionales juegan un papel primordial en lo que se conoce como *Inteligencia Ambiental (AmI)* [128].

Es así como, inmersas en este mundo, se encuentran omnipresentes la tecnología computacional (internet, computadores, portátiles, teléfonos inteligentes, tabletas digitales, sensores ambientales, localizadores basados en *GPS*, detectores de proximidad, lectores de tarjetas magnéticas, códigos de barra, identificadores automáticos mediante radiofrecuencia o *RFID*), las redes inalámbricas y los sistemas informáticos en lo que Weiser denominó *Computación Ubicua* o *Pervasiva (UbiComp)* en su artículo visionario publicado en 1991 [156]. La diferencia con respecto a dicha visión es que algunos de los dispositivos computacionales no son invisibles, aunque pueden ser muy discretos, y, por su uso continuo, las personas asumen estas tecnologías como parte de su vida normal.

Desde el punto de vista de sistemas automatizados hay tres factores fundamentales para el éxito de los mismos: el *hardware*, el *software* y las personas (*humanware*). Se puede tener la tecnología punta, tanto en equipos como en aplicaciones, pero si no se tienen las personas preparadas, debidamente entrenadas, ningún sistema funcionará adecuadamente. Es por ello que el *humanware* es considerado un factor crítico de éxito en estos tipos de sistemas.

Al centrarse en la perspectiva humana, surgen varias preguntas tales como ¿El aumento de la rapidez y eficiencia del *hardware* se traduce en mayor productividad del usuario? ¿Las facilidades aumentadas del *software* son amigables para el usuario? ¿La gran cantidad de dispositivos y redes (sensoriales

y/o inalámbricas) cada vez más presentes en la cotidianeidad mejoran la calidad de acceso a la información de las personas?

El reto actual de diferentes investigadores en el área de las TIC es hacer que los avances en computación (*hardware, software*, redes y dispositivos) mejoren la calidad de vida de las personas, porque normalmente se ha dejado de lado la perspectiva humana en los sistemas automatizados al privilegiar la eficiencia de los sistemas.

Si por un momento se analiza el ejemplo de una persona que debe consultar continuamente datos relacionados con su trabajo o vida personal, ¿la información presentada como respuesta a los usuarios está ajustada a sus requerimientos? En la mayoría de los casos sucede más bien lo contrario, la persona es abrumada por cantidad de información que no es de su interés y por lo general debe hacer esfuerzos mayores para filtrar las respuestas añadiendo nuevas restricciones a la consulta; mientras que una gran cantidad de sistemas no toman en cuenta para personalizar las respuestas datos importantes que pueden ser obtenidos a través de preferencias del usuario o por medio del ambiente.

Diversas áreas de Ciencias de la Computación proponen líneas de investigación en esta dirección. De acuerdo a Lieberman y Selker [83], la Inteligencia Artificial (AI) lo intenta haciendo a los computadores más inteligentes, mientras en el área de Interacción Persona-Ordenador (HCI) se buscan diseños más cuidadosos centrados en el usuario y con interfaces de manipulación directa [131]; esto definitivamente ayuda a la solución, pero para que realmente se logre el objetivo de mejorar la calidad de acceso de la información de las personas, se requiere abordar un problema importante en éstas y otras áreas de computación: ***la sensibilidad al contexto***.

Un significativo número de los fracasos del software de hoy en día que se manifiestan en mensajes de error, procedimientos tediosos y comportamiento inadecuado, son debidos a que el programador supuso ciertas asunciones y se realizan acciones que pueden ser correctas en un contexto fijo, pero no son las más adecuadas para todos los distintos contextos del usuario [83]. La manera de solucionarlo es que el software conozca más acerca del contexto, convirtiéndose en sensible (o consciente) al mismo. Hacer *lo adecuado* implica que lo sea dado el contexto actual del usuario.

Existen pocas aplicaciones a nivel de Sistemas de Información que tomen en cuenta de manera significativa el contexto. La mayoría actúan exactamente igual sin considerar cuándo, dónde y quién es el usuario; si es un principiante o un experto, si está solo o con amigos. Es por ello que nos cuestionamos *¿el usuario puede desear que el computador se comporte diferente bajo ciertas circunstancias?*

Una porción considerable de lo que actualmente se llama inteligencia en AI o buen diseño en HCI en realidad equivale a ser *sensible al contexto* en el cual los sistemas son utilizados. En el modelo tradicional de operación de un computador dada cierta entrada, se efectúa el proceso (sobre las entradas y/o información almacenada) y se obtiene la salida, con la certeza de que siempre que sea la misma entrada tendrá como resultado la misma salida. Frente a este modelo, se plantea un paradigma distinto de aplicaciones, donde a la entrada de datos se le suma el contexto de interacción con el sistema, se efectúa el proceso adaptado a ese contexto y se obtienen salidas que van a depender no sólo de los datos ingresados sino también del mencionado contexto.

Es por ello que se puede afirmar que se incluye cierto nivel de *incertidumbre* (imprecisión o vaguedad) debido a que no siempre las mismas entradas para una aplicación producen idéntica salida (manteniendo los datos sobre los cuales se consulta), porque el contexto afecta al ciclo básico computacional, al constituir información diferente a la proporcionada por los datos de entrada explícitos.

Además, existe otro factor que afecta a la calidad de respuesta de los sistemas ofrecidos a los usuarios, ya que los datos de entrada pueden ser afectados por cierto grado de imperfección que los sistemas de bases de datos tradicionales son incapaces de gestionar eficientemente. Esta imperfección está omnipresente en el mundo real, debido a que el usuario normalmente utiliza términos en el lenguaje natural que no pueden ser traducidos directamente al lenguaje tradicional que usan las máquinas. En estas aplicaciones es importante gestionar la imperfección de los datos adecuadamente, a fin de que los usuarios puedan tomar decisiones correctas y les sean proporcionados servicios de alta calidad. Y esta imprecisión también está afectada por el contexto.

La propuesta que se presenta en esta memoria toma en cuenta todos estos factores ya que las Bases de Datos Difusas proporcionan un marco teórico propicio para la gestión adecuada y eficiente de datos imperfectos. Tomando en cuenta que el lenguaje natural es ambiguo y la definición de ciertos términos dependerá de percepciones de la persona y/o aspectos contextuales, la sensibilización al contexto de las mismas permitirá desarrollar aplicaciones personalizadas y flexibles, las cuales proponemos sean de una forma amigable y lo más transparente que sea posible para el usuario.

En la presente investigación se propone un modelo para un Sistema de Bases de Datos Difusas Sensible al Contexto. Se analiza el concepto de contexto en estos sistemas y se propone una arquitectura general para hacerlos sensibles al contexto. Además, en el marco de esta propuesta, y como ejemplo de aplicación de la propuesta, se presenta un sistema gestor de bases de datos difuso en el que cada usuario puede establecer la semántica de los términos con los que se

comunica con el sistema, ingresando datos y obteniendo sus consultas de acuerdo a dicha semántica, a pesar de que en la bases de datos se almacenen los datos de acuerdo a un esquema prefijado por el diseñador del sistema; esto permite que un mismo término puede ser interpretado en forma diferente de acuerdo al lenguaje de cada usuario y su contexto.

Uno de los objetivos es sensibilizar al contexto aplicaciones de Sistemas de Información, utilizando Bases de Datos Difusas para representar valores que tradicionalmente son imperfectos, reflejando lo que sucede en el proceso de comunicación entre seres humanos, para conseguir acciones inteligentes. Hasta donde alcanza nuestro conocimiento, hay pocas investigaciones que aborden el problema del contexto utilizando bases de datos difusas, destacando algunas como las presentadas en [62] y [173].

La principal diferencia de estos enfoques con el modelo propuesto es que no se toma en cuenta la representación de datos imperfectos. En nuestra propuesta se plantea utilizar estructuras lógicas proporcionadas por los SGBDOR, permitiendo flexibilizar la las aplicaciones de una manera eficiente, evitando capas lógicas con software añadido, desarrollado en un lenguaje de programación no nativo del SGBDOR, sobre el gestor de bases de datos que manipulen y traduzcan las instrucciones. Además, al utilizar el estándar SQL:1999 para gestionar los datos imperfectos sólo se requiere que el usuario conozca dicho estándar.

Concretamente, se va a recurrir al modelo de dominios difusos propuesto por Cuevas et al. [41] [42], efectuando su implementación en el Gestor de Bases de Datos Objeto Relacional Oracle (SGBDOR) utilizando métodos de acuerdo a las definiciones de Marín et al. [90] [91]; aprovechando las capacidades objeto-relacionales del SGBDOR para extender, mediante un enfoque orientado a objetos, los tipos de datos. Además se presentará un modelo de contexto en esquemas de interacción de Sistemas de Bases de datos Sensibles al Contexto y una arquitectura que sirva de marco de trabajo para el desarrollo de este tipo de sistemas.

Por último, para demostrar la validez del modelo propuesto se ha implementado una prueba de concepto de una aplicación web con interfaz móvil en el área médica, permitiendo al usuario ingresar datos de acuerdo a su lenguaje particular y obtener consultas flexibles de acuerdo a su contexto. La aplicación fue desarrollada para el Hospital Ortopédico Infantil de Venezuela, enfocándonos en el Examen físico articular realizado en el Laboratorio de Análisis de la Marcha.

1.2. Objetivo General

De acuerdo a lo establecido en la introducción, el objetivo general de la tesis es proponer un modelo para el desarrollo de Sistemas de Bases de Datos Sensibles al Contexto, con el interés de que este tipo de sistema responda adaptativamente tomando en cuenta el contexto del usuario y utilizando representación de valores imperfectos basada en el enfoque proporcionado por las bases de datos difusas. En nuestra propuesta, personalizamos la experiencia del usuario sin abrumarlo con acciones excesivas, añadiendo nuevas prestaciones a los sistemas de bases de datos tradicionales, mediante la utilización de estándares de reconocido uso en la comunidad de bases de datos.

Hacemos un estudio profundo del contexto en diversas áreas de computación y cómo es utilizado para desarrollar aplicaciones sensibles al contexto, investigando cómo ha sido utilizado este concepto en la interacción del usuario con una base de datos, donde los resultados de las consultas puedan ser afectados por el contexto en que se hacen las mismas. Asimismo, tomamos en cuenta las experiencias obtenidas en el desarrollo de bases de datos difusas y cómo puede aplicarse la Teoría Computacional de Percepciones [168]. Diseñamos métodos dentro de un gestor de bases de datos objeto-relacional para que sean implementados por personas expertas, haciéndolos transparentes a usuarios finales inexpertos mediante el desarrollo de aplicaciones basadas en el modelo.

Queremos contribuir a desarrollar sistemas de información que mejoren la experiencia del usuario a través de la integración de los datos contextuales, con las siguientes ventajas: permitir la representación de datos imperfectos y la elaboración de consultas flexibles, reproducir en los sistemas de gestión de datos lo que sucede en el proceso de comunicación entre personas. Como una particularización del modelo propuesto presentamos la semántica adaptable al contexto de atributos lingüísticos. Por otra parte, se plantea el desarrollo de aplicaciones de propósito específico, amigables para usuarios inexpertos, donde se le da importancia a la utilidad de los resultados de la consulta.

La validez de la propuesta la demostramos a través de una prueba de concepto en el área médica para el Hospital Ortopédico Infantil de Venezuela. La implementación admite definir dependencias contextuales, permitiendo ingresar datos en forma personalizada y obtener consultas de acuerdo a la preferencia del usuario.

1.3. Objetivos Específicos

El cumplimiento del objetivo general se logra a través de los siguientes objetivos específicos:

- Realizar un estudio de los antecedentes de la utilización del Contexto en Computación y su incorporación en Sistemas de Información
- Definir el modelo de contexto en un esquema de interacción de un usuario con un Sistema de Bases de Datos Sensible al Contexto.
- Proponer una arquitectura para construir Sistemas de Bases de Datos sensibles al contexto.
- Desarrollar la anterior para su aplicación en sistemas con bases de datos difusas y proponer un modelo para la representación y manipulación de semántica adaptable en atributos lingüísticos.
- Demostrar mediante una prueba de concepto en un dominio de aplicación real la utilidad de las propuestas desarrolladas en esta memoria.

1.4. Metodología

La investigación ha de ser vista como un proceso sistemático de recolección y análisis lógico de información con un fin concreto. En cualquier caso, este proceso fundamentalmente se refiere a la búsqueda reflexiva, en la que cada decisión adoptada por el investigador se describe de manera explícita y se justifica racionalmente su elección. En esta investigación se estudia el contexto del usuario en el ámbito de sistemas de bases de datos difusas para contribuir en el desarrollo de sistemas inteligentes, flexibles y personalizados.

Es por ello que se efectúa una investigación documental, valiéndose principalmente de bases de datos electrónicas de reconocido prestigio en la comunidad científica del área de sistemas informáticos, para consultar trabajos e investigaciones hechas por otros autores, estudiando el problema abordado con el propósito de ampliar y profundizar el conocimiento de su naturaleza.

En cuanto al diseño de la investigación, que se refiere a la estrategia que adopta el investigador para responder al problema planteado en el estudio [63], es de carácter no experimental, ya que se realiza sin manipular deliberadamente variables independientes para ver su efecto sobre otras variables; lo que se hace es observar fenómenos tal como se dan en su contexto natural, para después analizarlos. En este tipo de estudio se observan escenarios ya existentes, no provocados intencionalmente por el investigador.

Con respecto al enfoque, la presente investigación responde a uno cualitativo que consiste en hacer ordenamientos conceptuales, considerando significados alternativos de los fenómenos observados [63], añadiendo dos características muy importantes: fueron sistemáticos y creativos al mismo tiempo.

Por otro lado, el grado de profundidad con que fue abordado el objeto de estudio fue proyectivo, dado que se proponen soluciones a una situación determinada [110]. Para llegar a estas soluciones se exploró, describió y explicó el modelo realizando una prueba de concepto para demostrar la validez del mismo. A través del desarrollo del modelo Bases de Datos Difusas Sensibles al Contexto se proponen mejoras en los sistemas informáticos actuales y se abren perspectivas para futuros trabajos.

En esta investigación se diseñó el modelo dirigido a cubrir una necesidad derivada de los retos impuestos debido a la creciente complejidad en la gestión de datos del entorno informático actual y basado en estudios anteriores. De allí que el término proyectivo se refiere a un proyecto en cuanto a aproximación o modelo teórico. El investigador llegó a esta propuesta por medios diferentes, los cuales involucraron procesos, enfoques, métodos y técnicas propias. El modelo teórico propuesto puede ser utilizado por aplicaciones que se basen en el uso del contexto en el ámbito de bases de datos difusas para el desarrollo de sistemas inteligentes, flexibles y personalizados.

En el presente estudio, luego de la revisión de antecedentes y análisis documental, se utilizaron técnicas de modelado orientado a objetos así como su implementación en un Gestor de Bases de datos Objeto Relacional tal como Oracle y se desarrollaron prototipos de aplicaciones sensibles al contexto. Se utilizó una estrategia de análisis de lo general a lo particular (*top-down*) y para el desarrollo de la prueba de concepto se utilizó una metodología iterativa e incremental, dado que se inició con una implementación en las bases de datos objeto relacional limitada y luego se fueron ampliando sus funcionalidades basadas en el modelo propuesto; posteriormente se desarrolló la prueba de concepto de una aplicación sensible al contexto limitada, la cual posteriormente fue ampliada de acuerdo a las necesidades de los usuarios.

1.5. Estructura del Trabajo

De acuerdo a los objetivos planteados, además de lo presentado en este capítulo, la tesis doctoral se organiza como sigue:

En el Capítulo 2 se presenta el estudio de los conceptos previos y antecedentes relacionados con el objeto de estudio. Se muestran definiciones del contexto en diversas áreas, se describen las aplicaciones sensibles al contexto y

cómo ha sido la integración del contexto en Sistemas de Información a través de modelos de datos contextuales y el procesamiento de consultas sensibles al contexto.

Luego, en el Capítulo 3, se estudian los conceptos fundamentales de bases de datos difusas, el papel protagonista que ha tenido la lógica difusa en la representación de datos imperfectos, las bases de datos relacionales difusas, bases de datos orientadas a objeto difusas y, para finalizar, el modelo de bases de datos de objetos difusos que constituye la base teórica para el modelo propuesto.

En el Capítulo 4, como parte de resultados de nuestro trabajo, se desarrolla el modelo formal de un Sistema de Bases de Datos Sensible al Contexto mediante una presentación del esquema de interacción del usuario con un sistema de bases de datos sensible al contexto, la definición y descripción del modelo de contexto propuesto en este tipo de ambientes, la sensibilidad al contexto de la salida del sistema, terminando con la presentación de la arquitectura de un Sistema de Bases de Datos Sensible al contexto, en un alto nivel de abstracción, que sirve como marco de desarrollo dentro del modelo que proponemos.

Posteriormente en el Capítulo 5, como un desarrollo particular de la validez del modelo propuesto, se muestra cómo hacer realidad la utilización de atributos lingüísticos con semántica adaptable al contexto, se describe el problema, un marco teórico incluyendo el modelo de dominios difusos y el problema de la semántica; luego se describe un Modelo Formal como otro resultado importante de nuestra tesis, para concluir con la explicación de cómo se implementa de forma transparente en un Sistema Gestor de Base de Datos Objeto Relacional.

En el Capítulo 6 se muestra la implementación de una prueba de concepto en un dominio del área médica, mediante la utilización del lenguaje estándar SQL:99 del Sistema Gestor de Bases de Datos Objeto-Relacional Oracle. También se describe una aplicación web con su interfaz móvil, a través de: diseño de la aplicación, interfaces, consultas y sesión de administrador. Por último, mostramos cómo es la utilización de la arquitectura propuesta en la aplicación del Examen Físico Articular del Hospital Ortopédico Infantil de Venezuela.

Finalmente, en el Capítulo 7, se presentan las conclusiones y líneas futuras de trabajo.

Capítulo 2

Estudio Preliminar

En este capítulo se presentan los fundamentos teóricos y los antecedentes sobre los que se basará nuestra propuesta. En primer lugar se estudia la definición de contexto y cómo se ha usado en diversos campos de la Ciencia de la Computación. Así como los diferentes tipos de modelo del contexto, importantes para definir el que se propondrá para su utilización en el desarrollo de Sistemas de Bases de Datos Sensibles al contexto.

El contexto puede ser producido por diversos factores, los cuales pueden ser agrupados a grandes rasgos en dos categorías principales: (a) el usuario (perfil o datos personales), (b) el entorno o ambiente (localización, tiempo, dispositivo de conexión); y se pueden producir datos contextuales proporcionados por otros agentes de software o paradigmas emergentes de computación (Internet de las cosas). Sin dejar de considerar este último aspecto, en nuestro estudio nos enfocaremos en las dos categorías principales.

Además, se analizan las aplicaciones sensibles al contexto, su utilización en Sistemas de Información y especialmente en la gestión de datos; destacando la necesidad de utilizar información contextual para el desarrollo de aplicaciones que resulten más satisfactorias para los usuarios.

2.1. El Contexto

El término se deriva del latín *contextus* y se refiere a todo aquello que rodea (física o simbólicamente) a un acontecimiento. El contexto implica un proceso activo relacionado con la manera en que los humanos enlazan su experiencia con el ambiente para darle un significado a las palabras o acciones. La noción de contexto ha sido investigada en áreas disímiles tales como Sociología, Psicología, Filosofía y Computación.

En Sociología, ciertas investigaciones han hecho hincapié en la importancia del contexto para observar el comportamiento de las personas y para

entender sus capacidades cognitivas [83]. En psicología las teorías de *acción* y *cognición situada* subrayan el efecto del contexto social sobre el comportamiento humano.

Fue Suchman [139], quien primero utilizó el término *acción situada*, estableciendo un paralelo entre las normas de la sociología funcionalista y los planes de la psicología cognitiva, presumiendo la existencia de una gran cantidad de antecedentes (*background*) de afirmaciones y presuposiciones a las cuales responden nuestras actitudes y acciones. La definición de acción situada sugiere que todo curso de acción depende esencialmente de las circunstancias materiales y sociales, por lo que esta teoría se centra en estudiar cómo las personas usan sus circunstancias para lograr *acciones inteligentes*.

Por otro lado, la teoría de la *cognición situada* enfatiza la importancia del medio ambiente como una parte integral del proceso cognitivo. Suchman ha hecho contribuciones fundamentales al análisis etnográfico, el análisis conversacional y técnicas de diseño participativo para el desarrollo de sistemas informáticos interactivos.

Autores como Kaptelinin [73] y Nardi [107] discuten la teoría de la actividad e investigan cómo los contextos de la conducta determinan lo que constituye un comportamiento exitoso y las estrategias que debe emplear un agente para generar dicho comportamiento.

El contexto, no sólo el físico sino el organizacional y el cultural, juega un rol esencial en la conformación de las acciones para proporcionar a las personas los medios de interpretar y comprender la acción. Esto es debido a que el significado de la acción es determinado interactivamente involucrando el contexto temporal, así como las acciones y expresiones adquieren su significado e inteligibilidad por lo que representan como parte de un patrón amplio de actividad.

La comunicación verbal es señalada por Suchman [139] como el prototipo de un enfoque contextual sobre la acción donde se pueden distinguir dos dimensiones, a saber:

- *Indexicalidad del lenguaje*, referido al hecho de que una expresión toma su significado de las circunstancias en las cuales es presupuesta o de aquello que es indicado o percibido, por lo que el lenguaje tiene una fuerte dependencia contextual del sujeto que lo explota.
- *Reflexividad de toda acción con propósitos*, el lenguaje situado no solo está enraizado en la situación, sino que en gran medida constituye su situación de uso, definiendo el marco de acción; es decir, el lenguaje crea y sostiene un entendimiento compartido en ocasiones específicas de interacción.

Así, cuando los seres humanos se comunican, el contexto actúa, en muchos casos, de manera implícita. Por ejemplo, la frase *está haciendo frío* puede significar temperaturas muy distintas dependiendo de si la persona está en un país nórdico o en uno tropical. En el caso de la frase: *quiero escoger una persona joven*, el significado de la edad cambia notablemente cuando el objetivo es escoger a un gimnasta o a un profesor. Cuando se dice: *necesito ir a cenar a un restaurante económico*, la palabra económico puede interpretarse de manera diferente, incluso para una misma persona, si el motivo de la cena es para un encuentro romántico, la cena es entre compañeros de trabajo o la persona tiene poco dinero en ese momento; aquí el significado del término puede estar dado por motivos emocionales o por disponibilidad de recursos económicos, entre otros.

Estos trabajos mencionados son fundamentales en el desarrollo de la Interacción Persona-Ordenador y ofrecen una base para entender nuestra relación con la tecnología. El contexto tiene un impacto significativo en la manera en que las personas actúan y en cómo interpretan las cosas; en otras palabras, un cambio en el contexto produce una transformación en la experiencia que se está viviendo, es por ello que se critica que la mayoría de ingenieros de sistemas efectúan sus diseños independientes del contexto y que las aplicaciones sean rígidas, es decir, que no logren adaptarse en forma dinámica a la situación. En la próxima subsección abordamos el contexto enfocándonos más en su uso en el área de Computación

2.1.1. El Contexto en Computación

En Computación Ubicua (*UbiComp*), se refiere al contexto como información que puede ser usada para caracterizar la situación de una entidad (persona, lugar u objeto) considerada relevante en la interacción entre el usuario y una aplicación. Términos comúnmente utilizados en esta área son: *contexto del usuario* (perfil, ubicación), *contexto físico* (niveles de ruido, temperatura) y *tiempo*.

En cuanto al enfoque dentro de UbiComp, éste se orienta principalmente a gestionar el contexto proporcionado por sensores, que puede ser automáticamente adquirido en un ambiente físico y tratado como entrada implícita para afectar positivamente el comportamiento de una aplicación. En este sentido, se considera normalmente como contexto a la ubicación, identidad y estado tanto de las personas, como la de grupos y objetos (físicos y computacionales).

Baldauf, Dustdar y Rosenberg [11] recopilan definiciones de contexto en UbiComp de varios investigadores las cuales se resumen a continuación.

- Schilit y Theimer (1994): localización, identidades de personas cercanas, objetos y cambios a dichos objetos. En la literatura son los primeros que introducen el término sensibilidad al contexto (*context-aware*).
- Ryan et al. (1997): ubicación del usuario, ambiente, identidad y tiempo.
- Dey (1998): estado emocional del usuario, centro de atención, ubicación y orientación, fecha y hora, así como los objetos y personas en el ambiente del usuario.
- Hull et al. (1997): los aspectos de la situación actual.

Definiciones intensivas más enfocadas al concepto son mostradas a continuación.

- Brown (1996): los elementos del ambiente del usuario los cuales son conocidos por el ordenador.
- Dey y Abowd (2000): cualquier información que puede ser usada para caracterizar la situación de entidades (esto es, si es una persona, lugar u objeto) que es considerada relevante a la interacción entre un usuario y una aplicación, incluyendo tanto al usuario como a la aplicación.

Un rol más profundo de la definición abordada en esta sección, es el sugerido por Dourish [50] en el ámbito de la Interacción Persona-Ordenador (HCI), quien opina que el contexto en el que las acciones tienen lugar es lo que permite a las personas encontrarlas significativas. Afirma el autor mencionado que, además de la visión de Computación Ubicua establecida por Weiser en 1991 [156], hay que tomar en cuenta las investigaciones en interfaces tangibles propuestas por Ishii y Ullmer's [68], así como los trabajos sociológicos de la organización del comportamiento interactivo que estudió estos aspectos bajo el enfoque de la fenomenología filosófica. Su propuesta es que deben integrarse todas estas perspectivas para entender los fundamentos que las unen y proporcionar un marco compartido sobre los principios básicos detrás de diversas formas de interacción *personalizadas*.

También en el área de HCI, una definición de contexto dada por Chen, Finin y. Joshi [37] es: cualquier información que puede ser utilizada para caracterizar la situación de una entidad (persona, dispositivo computacional u objeto físico no computacional) considerada relevante en la interacción entre un usuario y una aplicación, incluyendo actividades y roles. Por ejemplo, en un ambiente de "Sala de conferencia inteligente", los contextos son:

1. La identidad de las personas y dispositivos,
2. La ubicación de las personas y dispositivos,
3. Las actividades en que las personas participan, y
4. Los roles e intenciones de personas que participan en las actividades.

Esta conceptualización es muy parecida a la presentada por Dey, Abowd y Salber [49]. Ellos afirman que la información en el ambiente físico y electrónico crea un *contexto* para la interacción entre humanos y servicios computacionales. Así, definen contexto como cualquier información que caracteriza una situación relacionada a la interacción entre usuarios, aplicaciones y el ambiente que los rodea.

Afirman Baldauf, Dustdar y Rosenberg [11] que una forma común de clasificar instancias del contexto es la distinción de diferentes *dimensiones contextuales*, las cuales se pueden resumir en dos grupos, a saber:

1. *Dimensiones contextuales externas (físicas)*: referida a lo medido por sensores de hardware tal como ubicación, iluminación, sonido, movimiento, temperatura o presión de aire.
2. *Dimensiones contextuales internas (lógicas)*: que es especificada principalmente por el usuario o capturada evaluando las interacciones del usuario tal como objetivos, perfil, actividades, procesos de negocio, estado emocional.

Aunque en HCI se utilizan ambos grupos de dimensiones contextuales, en UbiComp se hace uso sólo de las dimensiones contextuales externas, las cuales son fáciles de capturar mediante el uso tecnologías de detección fuera de la plataforma (*off-the-shelf*).

Cuando se trabaja con contexto, tres entidades pueden ser distinguidas: lugares (salones, edificios), personas (individuales, grupos) y cosas (objetos físicos, componentes computacionales) [49]. Cada una de estas entidades puede ser descrita por diversos atributos los cuales se pueden clasificar en cuatro categorías:

1. *Identidad*: cada entidad tiene un único identificador
2. *Ubicación*: una posición de la entidad, ubicación con respecto a otros objetos (*co-ubicacion*) o proximidad.
3. *Estado* (o actividad): significando las propiedades intrínsecas de una entidad, por ejemplo, temperatura y alumbrado de un salón, procesos en ejecución en un dispositivo.
4. *Tiempo*: usando marcas de tiempo (*timestamps*) para definir precisamente una situación u ordenando acontecimientos.

Un señalamiento importante lo hacen Chen, Finin y Joshi [37], quienes indican que los seres humanos aprovechan frecuentemente el contexto cuando se comunican y ejecutan acciones; debido a que son seres conscientes del contexto, capaces de transmitir ideas efectivamente, sin necesidad de establecer explícitamente información de contexto; aquí se destaca que las personas, a pesar

de poder ejecutar diversas tareas, efectúan actividades con un rol específico que determina el contexto.

En este orden de ideas Chen y Kotz [38], definen al contexto como el conjunto de estados del medio ambiente y configuraciones que, o bien son determinadas por el comportamiento de una aplicación, o en el que un evento de una aplicación se produce y es interesante para el usuario. Los autores clasifican al primer tipo de contexto como *activo*, y el segundo tipo como *pasivo*, lo cual es relevante pero no crítico para una aplicación. Esta definición se usa en utilización de dispositivos móviles conectados principalmente a través de redes inalámbricas.

En un aporte de robótica asistida, los autores Liu, Zhang y Li [84] afirman que usualmente se usa el contexto específico para la comprensión de la intención del usuario con un involucramiento mínimo del mismo. Especialmente en el caso de asistencia a mayores en sus actividades de vida diaria, el reconocimiento de la intención implícita puede proporcionarles una gran comodidad, además de mejorarles la calidad de vida.

Después de analizar las diferentes definiciones de contexto, en HCI se toma en cuenta al contexto desde el punto de vista de lo que las personas puedan encontrar significativo en la interacción con las aplicaciones, permitiendo hacer más productivas y efectivas las actividades que rodean a los usuarios con respecto a los ordenadores, con el objetivo de incrementar la *satisfacción* de los mismos, la *usabilidad* y *amigabilidad*. Esta definición de contexto se ajusta mejor al enfoque de nuestra propuesta.

También hay que destacar la existencia de aportes que integran el contexto en el área de computación no sólo en el área de UbiComp [11] y HCI [14] [49] [50] [64] [65], sino también en Robótica [84], Computación móvil [38] [81], Internet de las cosas [112].

2.1.2. Ideas generales acerca del Contexto en Sistemas de Información y Bases de Datos

En Sistemas de Información, podemos considerar un escenario general donde:

- Diferentes clases de usuarios acceden a variedad de información proveniente de múltiples fuentes de datos (internas y/o externas), a través de una o diversas aplicaciones;
- Se utilizan sensores para monitorizar la información proporcionada por el entorno almacenado en conjuntos de datos (*datasets*) de cualquier formato;

- El conocimiento del contexto se utiliza para controlar el flujo de información que llega a los usuarios, excluyendo la que está fuera del contexto (información ruido);
- Un aspecto fundamental de la gestión de contexto es hacer en forma transparente las operaciones de manipulación de datos utilizados para calcular y presentar la información personalizada para el usuario e independiente de la estructura lógica de la fuente de información.

Con estas premisas, Tanca et al. [141] afirman que el proceso de personalización de la información puede ser descrito principalmente por las siguientes funcionalidades:

- a) *Gestión del Contexto*: la función del diseñador es prever los contextos posibles a ser experimentados por el usuario y su aplicación durante la vida del sistema. El esquema de contexto utilizado como base del proceso de personalización, junto con el diseño de datos u operaciones sensibles al contexto, debe ser flexible, ya que los requerimientos del sistema sensible al contexto son dinámicos y pueden evolucionar.
- b) *Gestión de la personalización sensible al contexto*: el contexto actualmente activo debe ser reconocido. Algunos de los aspectos que caracterizan el contexto actual pueden ser automáticamente derivados del ambiente (localización y tiempo, rol del usuario en la aplicación o el tipo de dispositivo utilizado para acceder los datos), mientras que otros requieren una especificación explícita por parte del usuario o de la aplicación con capacidad de aprendizaje (por ejemplo, tópicos de interés del usuario). Una vez que el contexto es reconocido, la contextualización debe ser aplicada a las consultas del usuario, y los datos sensibles al contexto deben ser enviados a los usuarios y aplicaciones. Los intereses y preferencias de un usuario pueden variar de acuerdo al contexto actual del usuario, por lo que un cambio de contexto puede cambiar la importancia relativa de la información.
- c) *Gestión de acceso a los datos*: La proliferación de sitios web de datos intensivos (sitios que sirven grandes volúmenes de páginas cuyo contenido se extrae dinámicamente a partir de una bases de datos), así como iniciativas de datos enlazados (*linked data*) en la web accedidos libremente, proporcionan a los usuarios fuentes de información apreciables. Tales fuentes de datos heterogéneas, dinámicas y posiblemente móviles deben ser consultadas de una manera sensible al contexto. Para reconocer el contexto actual se deben detectar los datos contextuales del ambiente utilizando sensores, constituyendo ésta, otra fuente de datos posible con la que se deba tratar.

En el área de Bases de Datos, merece especial mención el trabajo de Stefanidis, Koutrika y Pitoura [137] donde se definen preferencias contextuales como un par (C,P), donde C define el contexto y P la preferencia; la idea es que las preferencias sean condicionadas por circunstancias específicas (contexto). Por ejemplo, una persona puede preferir a un director de película dependiendo del género de la película. Aunque la idea de este trabajo en relación al contexto es muy limitada, de esta investigación tomaremos los aportes para la gestión de datos contextuales en una manera procesable por la computadora.

Otro aporte significativo lo presentan Bolchini et al. [18] quienes proponen CARVE (*Context-Aware Relational View dEfinition*), una metodología para definición de vistas sensibles al contexto, la cual está diseñada para aplicaciones dinámicas y modernas que, en diferentes ambientes y situaciones, necesitan acceder a diferentes porciones de datos. Aunque en nuestra propuesta no se utilizan vistas sensibles al contexto, es de nuestro interés el modelado contextual propuesto por estos autores para definir los datos contextuales.

Por último es de hacer notar que existen varias propuestas para integrar el contexto en Sistemas de Información que pueden ser consultadas en [6] [37] [132] [141] y específicamente en el área de Bases de Datos [8] [18] [62] [95] [101] [109] [124] [137]. En la próxima sección trataremos aspectos fundamentales que tomamos en cuenta para integrarlos en nuestra propuesta.

2.2. Aplicaciones Sensibles al Contexto

Una vez analizada la idea de contexto vamos a analizar qué debe hacer una aplicación sensible al contexto, cómo se comportan y qué tipos existen.

2.2.1. Consideraciones Generales

Tradicionalmente, en la teoría de Ciencias de la Computación se ha tomado una posición diametralmente opuesta al problema del contexto, es decir, tener sistemas informáticos independientes del mismo. Las funciones, predicados, subrutinas, sistemas de entrada/salida y redes, normalmente son abstracciones matemáticas tratadas como cajas negras, basados en una idea de contexto fija y determinada por el diseñador del sistema. Algo entra en un lado y algo sale del otro lado, donde la salida es completamente determinada por la entrada.

Al contrario, se dice que una aplicación es sensible al contexto, cuando puede decidir qué hacer, basándose no sólo en la entrada explícitamente presentada, sino también en el contexto; además, el resultado puede afectar no

sólo a la salida explícita sino igualmente al contexto [83], por ejemplo un usuario puede estar modificando datos de su perfil que es parte de su contexto.

En la figura 1 se puede observar que contexto en computación se considera todo lo que afecte al ciclo computacional, excepto las entradas y salidas explícitas, por lo que aplicaciones capaces de obtener y modelar del mundo externo aspectos relacionados a un evento se consideran *sensibles al contexto*.

Tómese en cuenta que este es un modelo simplificado, porque hoy en día este ciclo incluye la realimentación de la entrada de acuerdo a las salidas y el paralelismo en los procesos, es decir, pueden existir múltiples operaciones para conseguir una salida específica.

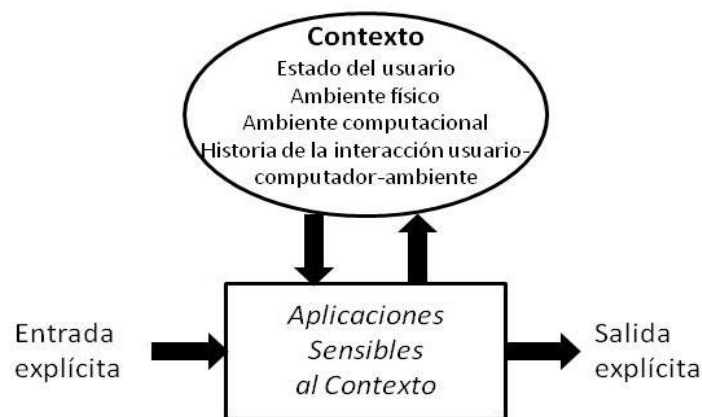


Figura 1. El contexto y el ciclo computacional, tomado de Lieberman y Selker [83]

El concepto de sensibilización al contexto (*context-awareness*) utilizado en diversas áreas como Computación Ubicua, Sistemas de Recomendación y Recuperación de Información, normalmente se orienta a construir un perfil del usuario, el cual puede ser hecho de forma explícita (mediante un cuestionario al usuario) o implícita (evaluando las interacciones del usuario con las aplicaciones para definir dicho perfil).

La acción de un usuario es efectuada en un cierto ambiente; es por ello que la interacción Persona-Ordenador *implícita* es la acción efectuada por el usuario que no está principalmente destinada a interactuar con un sistema informático, pero que tal sistema la entiende como entrada [131]. La interacción implícita está basada en la asunción de que el ordenador tiene un cierto entendimiento del comportamiento de la persona en una situación dada. Este conocimiento es entonces considerado como una entrada adicional al ordenador mientras hace una tarea. Si se buscan los conceptos necesarios para facilitar la interacción implícita, se pueden identificar tres bloques básicos de construcción:

1. La habilidad de tener percepción del uso, el ambiente y las circunstancias.
2. Mecanismos para entender lo que los sensores ven, oyen y sienten.
3. Aplicaciones que pueden hacer uso de esta información.

A nivel conceptual los bloques 1 y 2 pueden ser descritos como contexto situacional y el 3 como aplicaciones sensibles al contexto. La noción de contexto es usada de diferentes formas; Schmidt [131] propone considerar al *contexto situacional* para describir fragmentos de interacción implícita, extendiendo el concepto de contexto más allá del contexto de información en entornos del mundo real. Se identifica la percepción e interpretación del usuario, el ambiente y las circunstancias como conceptos claves para la interacción implícita; además se requieren de aplicaciones que aprovechen esta información.

Uno de los objetivos, cuando se trata de la adquisición del contexto, es determinar lo que el usuario está procurando efectuar. Debido a lo difícil que es el hacerlo directamente, se pueden utilizar pistas dadas por el contexto para inferir esta información y notificar a una aplicación la mejor forma de apoyar al usuario. En el mundo real, la mayoría de características del contexto no pueden ser detectadas en forma automática y las aplicaciones deben confiar en lo que el usuario proporcione explícitamente, aunque esto supone una mayor carga para el mismo y cierto grado de incertidumbre si no es capaz de transmitirlo adecuadamente.

Por otra parte la sensibilidad al contexto ha constituido parte integral de ciertas aplicaciones informáticas. Aún formas simples de contexto, tal como el tiempo y la identidad, han sido usadas en forma significativa. A manera de ejemplo, se tiene que hay software de gestión personal que puede recordarnos eventos marcados en un calendario al ser sensibles al tiempo (hora y fecha) actual. El sistema operativo puede personalizar la apariencia de la interfaz al conocer la identidad del usuario a través del inicio de sesión (*login*).

Diferentes tipos de contexto pueden ser utilizados en conjunto. Por ejemplo, los sistemas pueden etiquetar archivos con el tiempo y la identidad, proporcionando diversas formas de organizar y encontrar información creada en el pasado.

A pesar de los avances mostrados en las TIC y el aumento en el poder computacional del hardware (incluyendo los dispositivos móviles), no se ha explotado suficientemente la noción de contexto en el procesamiento de conocimiento para el desarrollo de sistemas sensibles al contexto. En este respecto, Orsi y Tanca [109] proponen el uso de *Datalog* para considerar el contexto como un factor destacado en el diseño de sistemas y la gestión de actividades.

Ya más enfocado a sistemas de Bases de Datos los autores Pitoura, Stefanidis y Zaslavsky [116] afirman que una aplicación sensible al contexto puede apoyar en una de las siguientes características:

- Presentación de información y servicios a los usuarios
- Ejecución automática de un servicio para un usuario, y
- Anotación de la información con el contexto para apoyar próximas consultas.

Una vez establecida la definición de aplicaciones sensibles al contexto, en la próxima sección se estudia su categorización tomando en cuenta diferentes aspectos.

2.2.2. Comportamiento de las Aplicaciones Sensibles al Contexto

Los avances en miniaturización, redes inalámbricas y tecnologías de sensores han habilitado a los ordenadores para ser usados en más lugares y tener una mayor sensibilidad al mundo dinámico del que son parte. Durante los últimos años ha sido desarrollada una nueva clase de aplicaciones sensibles al contexto que hacen uso de estas tecnologías, mostrando cómo los ordenadores pueden aprovechar incluso las nociones elementales de ubicación, identidad, proximidad y actividad con gran efecto. La clave de estas aplicaciones es que proporcionan fuertes lazos entre los mundos físicos y sociales donde convivimos con el mundo virtual en el cual operan los ordenadores.

Es así como en los sistemas informáticos actuales el contexto no sólo es considerado un estado (como tiempo o ubicación) sino como parte de un proceso donde está involucrado el usuario [109], por lo que se han diseñado aplicaciones más sofisticadas sensibles al contexto que gestionan:

- Comunicación: capacidad de adaptar presentación de contenidos a diferentes canales o a diferentes dispositivos (sistema de comunicación con los usuarios)
- Sensibilización situacional
 - Modelar aspectos de ubicación y ambiente (*situación física*)
 - Modelar lo que está haciendo el usuario en el momento (*situación personal*)
 - Hacer implícita la interacción del usuario (*adaptar* la información a las necesidades del usuario)
- Gestionar fragmentos de conocimiento (*chunks of knowledge*), determinando el conjunto de informaciones, servicios o comportamientos relevantes a aplicaciones/situaciones.

Según Schilit, Adam y Wan citados por Pitoura, Stefanidis y Zaslavsky [116], existen varios tipos de aplicaciones sensibles al contexto dependiendo de la manera en que es utilizado el contexto, entre las que están:

- *Selección de objetos cercanos* es una técnica de interfaz de usuario donde se destacan los objetos que se localizan cerca o se hacen más fácil de escoger.
- *Reconfiguración contextual automática* es el proceso de añadir nuevos componentes, eliminar existentes o alterar las conexiones entre componentes debido a cambios en el contexto. Componentes típicos y conexiones son servidores y sus canales de comunicación con clientes. Sin embargo componentes reconfigurables también pueden incluir controladores de dispositivos, módulos de programas y elementos de hardware.
- *Información contextual y comandos*, pueden producir resultados diferentes de acuerdo al contexto en el cual son ejecutados.
- *Acciones disparadas por el contexto* son simples reglas *IF-THEN* usadas para especificar cómo podrían adaptarse sistemas sensibles al contexto. Son similares a información contextual y comandos, excepto que éstas son invocadas automáticamente.

De acuerdo a la primera categorización, en nuestro modelo aplicaremos *sensibilización situacional*, debido a que se modela la situación física y situación personal, además de adaptar la información a las necesidades del usuario; y con respecto a la segunda clasificación desarrollamos un tipo de aplicación de *información contextual y comandos*, ya que se producen resultados diferentes de acuerdo al contexto en el cual son ejecutadas las consultas.

Uno de los aspectos más abordados en investigaciones que se refieren a los principios para el desarrollo de aplicaciones de sensibles al contexto es la gestión del contexto [65] en los Sistemas de Información, tema que será tratado en la próxima sección.

2.3. Sistemas de Información Sensibles al Contexto

Una vez definido el contexto en Computación y las aplicaciones sensibles al contexto, nos enfocaremos en la revisión de conceptos en el área de Sistemas de Información específicamente ya que es de nuestro interés para el desarrollo del modelo propuesto.

2.3.1. Ideas Generales

Las características del contexto lo convierten en una tarea difícil de gestionar para los sistemas de información sensibles al contexto. Estas características incluyen las siguientes [116]:

- La información contextual es imperfecta.
- Algunos tipos de contexto (por ejemplo, perfil del usuario) son relativamente estáticos, mientras que otros (tal como ubicación) son dinámicos.
- Hay muchas alternativas de presentaciones del contexto ofreciendo variación en detalles y profundidad
- Hay complejas dependencias entre los parámetros contextuales, que algunas veces son difíciles de deducir.

Para la gestión del contexto, Martinenghi y Torlone [95] proponen la extensión del modelo de bases de datos relacional, con nuevos operadores de algebra relacional y un lenguaje de consulta extendido, que consideren el contexto para el desarrollo de sistemas de bases de datos sensibles al contexto. Aquí el contexto es tomado en cuenta para filtrar resultados ya sea en tuplas o en atributos de las relaciones.

Por su parte, Roussos y Sellis [126] abordan el problema de seleccionar los datos más apropiados de acuerdo a un contexto de referencia y en particular, en las respuestas a consultas sensibles al contexto. Estos autores proponen un modelo conceptual de datos basado en la definición de relaciones multi-esquema, además de un conjunto de operaciones que extienden el álgebra relacional para incorporar la noción de gestión de datos sensibles al contexto en el modelo relacional.

Aparte de las dos propuestas anteriores, que están en el marco de la incorporación de la gestión de datos contextuales en el modelo de datos relacional, hay otras que gestionan los datos contextuales a través de capas lógicas sobre el SGBD tales como Bunningen, Feng, y Apers [30]; y, Hadjali, Mokhtari y Pivert [62]. Mientras que Miele et al. [103] proponen la definición de vistas de usuario de acuerdo a cada contexto y Levandoski et al. [82] lo hacen a través de la incorporación de nuevas funcionalidades en el procesador de consultas del SGBD.

Una de las debilidades de estas propuestas es que las dos primeras pueden afectar el rendimiento al incorporar capas lógicas sobre el SGBD como mediadores utilizando lenguajes no nativos, mientras que todas incorporan lenguajes añadidos o extensiones no estándares, aumentando la complejidad para el usuario, siendo intrusivos y poco amigables para el mismo.

Luego de establecer las ideas generales del contexto en Sistemas de Información, debido a la importancia para nuestra propuesta seguidamente trataremos el tema del modelado de datos contextuales.

2.3.2. Procesamiento de Consultas Sensibles al Contexto

Pitoura, Stefanidis y Zaslavsky [116] indican que, aunque para la fecha de su informe, existían algunas investigaciones en procesamiento de consultas sensibles a la ubicación, la integración de otras formas de sensibilización al contexto al procesamiento de consultas es un tema tratado sólo en el marco de inteligencia ambiental por autores como Bunningen, Feng y Apers [30]. Más recientemente pero centrándose en preferencias contextuales lo han hecho Miele et al. [102]; Hadjali, Mokhtari y Pivert [62]; y, Levandoski et al. [82]; de éstas, las dos primeras abordan el problema de cómo inferir las mismas.

El procesamiento de consultas sensibles al contexto comprende varios aspectos, entre los que destacan: a) Los resultados de la consulta, b) optimización de la consulta, y c) la manera en que los resultados son presentados a los usuarios [116]. A continuación se explican cada uno de ellos.

a) Consultas con resultados sensibles al contexto

El contexto puede afectar los resultados producidos por una consulta. En este caso, la misma consulta puede producir diferentes resultados dependiendo del contexto en la cual es ejecutada. El procesamiento de una consulta sensible al contexto puede verse como un proceso de dos pasos, donde el contexto relevante a una consulta primero es identificado y adquirido, y luego es integrado en la consulta.

Las formas de incorporar el contexto en el procesamiento de consultas que proponen Pitoura, Stefanidis y Zaslavsky [116] son:

1. Permitir acceso explícito a parámetros contextuales dentro de una consulta, utilizando el contexto como una tabla más en consultas a bases de datos relacionales.
2. Aumentar la consulta con predicados añadiendo nuevas restricciones que utilicen atributos contextuales.
3. Definir preferencias contextuales para que la consulta devuelva respuestas con mayor relevancia según las condiciones. Así sobre un contexto específico se prefieren unas respuestas sobre otras.
4. Usar el contexto para el recuerdo asociativo de eventos pasados. Para ello los objetos o hechos pasados deben estar asociados al contexto en que sucedieron.

b) Optimización de consultas sensibles al contexto

La información de contexto puede usarse para derivar planes de ejecución con costos más efectivos, por ejemplo produciendo primero los resultados más relevantes (de acuerdo al perfil del usuario) o realizar la optimización tomando en cuenta parámetros diferentes a los normales (dar prioridad a métricas tales como energía utilizada por el ordenador en lugar de número de accesos a disco).

c) Presentación de las respuestas a consultas sensibles al contexto

La forma en que los resultados son presentados a un usuario puede depender de aspectos del contexto tales como el tipo de dispositivo que está utilizando el usuario para hacer la consulta y donde se van a desplegar los resultados. Además, consideraciones de energía y velocidad de conexión a la red pueden afectar la manera en que los resultados son mostrados al usuario (textual, sonido o multimedia).

De acuerdo a lo visto hasta ahora las principales propuestas de incorporación del contexto en sistemas de bases de datos se enfocan en el problema de preferencias contextuales, donde el contexto es un aspecto adicional en la gestión de preferencias como problema principal. Nuestra propuesta aborda el problema de la rigidez de las consultas en sistemas de bases de datos mediante la incorporación del contexto con un papel protagonista.

En la próxima sección revisamos algunos aportes interesantes de la revisión hecha en este capítulo que tomaremos como base para desarrollar nuestra propuesta.

2.4. Modelado del Contexto

Tras haber estudiado diversas definiciones de contexto, cómo las aplicaciones pueden sensibilizarse al mismo y la necesidad de gestionar el contexto en Sistemas de Información, ahora hay que describir los datos contextuales, las interrelaciones, la semántica y las limitaciones o restricciones impuestas por la situación. Las herramientas conceptuales se plantean como elemento muy útil para su propósito. Estas herramientas permiten al usuario definir *qué* datos desea almacenar sin importarle el *cómo* se almacenan, es decir, las estructuras lógicas y su contexto.

En general un modelo de datos es una representación del mundo real que conserva sólo los detalles relevantes (poder de abstracción). Los procesos de definición del diseño conceptual exigen identificar los requerimientos de la

información de los usuarios y representar éstos en un modelo bien definido; para llevar a cabo esto se necesita observar cuidadosamente la naturaleza de las condiciones de los usuarios y el significado de la representación lógica de los mismos. A continuación se explican diferentes modelos contextuales encontrados en la literatura.

Debido a que los sistemas sensibles al contexto están presentes en nuestra vida cotidiana el modelado contextual se ha convertido en un tema relevante y un campo de investigación en expansión, se necesita de este tipo de modelo para definir los datos contextuales a gestionar en una manera procesable por la máquina [17].

Strang y Linnhoff-Popien [138] analizan las técnicas más relevantes para modelar el contexto en UbiComp, basado en las estructuras de datos utilizadas para representar e intercambiar información contextual en cualquier sistema sensible al contexto. Al utilizar estos modelos, su representación, lenguajes de consulta y algoritmos de razonamiento, se facilita la interoperabilidad de las aplicaciones.

Los sistemas de Computación Ubicua demandan de ciertas características de los modelos de representación contextuales para satisfacer los requerimientos que se mencionan a continuación.

- *Composición distribuida*: en descripciones contextuales particulares se carece de una instancia central que sea responsable para la creación, despliegue y mantenimiento de datos y servicios. La composición, administración y los datos del modelo contextual varían con una dinámica alta en términos de tiempo, topología de red y fuentes.
- *Validación parcial*: Es deseable validar parcialmente el conocimiento contextual sobre estructuras y a nivel de instancia versus un modelo contextual en uso, aún si no hay lugares o puntos en el tiempo donde el conocimiento contextual esté disponible en un nodo como resultado de la composición distribuida. Este aspecto es importante debido a la complejidad de las interrelaciones contextuales, las cuales hacen que los modelos sean modificados permanentemente de acuerdo a procesos de realimentación con los usuarios.
- *Riqueza y calidad de la información*: La calidad de la información suministrada por un sensor varía en el tiempo, además de la variación de la riqueza de información proporcionada por diferentes tipos de sensores que caracterizan una entidad en un ambiente de Computación Ubicua, por lo que un modelo contextual en estos ambientes debe soportar indicadores de calidad y riqueza.
- *Nivel de formalidad*: Es un reto describir hechos contextuales e interrelaciones mediante una forma precisa y trazable. Por ejemplo,

para ejecutar la tarea “imprimir un documento en una impresora cerca de mí”, es requerido tener una definición precisa de los términos usados en la tarea, por ejemplo qué significa “cerca” para “mí”. Es deseable que cada participante que interactúa en un ambiente de Computación Ubicua comparta la interpretación y el significado de los datos que se utilizan (entendimiento compartido).

- *Aplicabilidad para ambientes existentes*: Desde la perspectiva de implementación es importante que un modelo contextual sea aplicable dentro de la infraestructura existente en ambientes de Computación Ubicua, por ejemplo una arquitectura tal como servicios Web.
- *Incompletitud y ambigüedad*: El conjunto de la información contextual que caracteriza entidades, en un ambiente ubicuo, disponible en algún punto del tiempo usualmente es incompleta y/o ambigua, particularmente si es recolectada a través de redes sensoriales, es por ello que el modelo deberá contemplar este aspecto.

Algunos de los requerimientos descritos son abordados dentro de ciertas técnicas de modelado contextual, otros dentro de sistemas de razonamiento asociados. Es de sumo interés, para sistemas que se desarrollen en ambientes de Computación Ubicua (*UbiComp*), que se cumplan todos estos requerimientos al desarrollar un modelo contextual [138]. De acuerdo a los autores los modelos orientados a objetos y basados en ontologías son los más adecuados para cumplir con todos estos requerimientos.

Nuestra propuesta va a estar enmarcada en la utilización de un modelo de datos difuso orientado a objetos, es decir, se va a utilizar todas las ventajas de un modelado de datos orientado a objetos, añadiendo mayor riqueza semántica, poder de personalización y flexibilización proporcionada por la lógica difusa y conjuntos difusos dirigido a aplicaciones realizadas en el ámbito de sistemas de información, lo cual será objeto de estudio en el próximo capítulo. En la próxima sección estudiaremos cómo se puede incorporar la sensibilidad al contexto en sistemas de bases de datos, específicamente en el módulo de procesamiento de consultas.

2.5. Discusión

La perspectiva humana es un punto de gran importancia que tomaremos en cuenta en nuestra propuesta debido a que se desean desarrollar sistemas que apoyen más el *humanware*, permitiendo mejorar la calidad de vida de las personas [15]. En la propuesta se desea privilegiar la utilidad de los resultados de la consulta, desarrollando sistemas personalizados, tratando de asemejar lo que

sucede en el proceso de comunicación entre humanos, que se ajusten más a las necesidades del usuario y que acerquen el mundo virtual al real de los usuarios.

El proceso de comunicación entre las personas es considerado un modelo de un enfoque contextual de la acción. Los seres humanos dan significado a las palabras de acuerdo a las circunstancias que rodean la situación en que se expresan, además este significado puede estar influenciado por su situación, objetivos y necesidades, entre otros. En la mayoría de los sistemas informáticos este proceso de comunicación no se refleja en forma adecuada porque no se toma en cuenta el contexto proporcionado por el ambiente, dejando muchas veces de lado información importante que puede ser aportada por el contexto.

Con respecto al modelado contextual, los estudios demuestran que aquellos que intentan servir para todas las aplicaciones, al final resultan ser poco efectivos; la aplicabilidad práctica y la usabilidad, son parámetros importantes e inversamente proporcionales a la generalidad del modelo. Debido a la complejidad del problema y la gran variedad de aplicaciones los modelos contextuales, aunque tengan cierto grado de generalidad, deben estar centrados en un sub-problema específico de contexto. En particular nuestra propuesta se enfoca en el esquema de interacción de los usuarios con sistemas de bases de datos.

En cuanto al manejo de la incompletitud e imperfección de los datos contextuales que se desprende del estudio de modelos de datos contextuales, es importante señalar que, de acuerdo a nuestro conocimiento, las Bases de Datos Difusas son propicias para almacenar y manipular este tipo de datos, proporcionando las estructuras lógicas y herramientas conceptuales para representar datos imperfectos basados en un fuerte basamento teórico de lógica y conjuntos difusos.

Como se ha visto en áreas como la Computación Ubicua se ha trabajado en tomar información de contexto proporcionado por sensores ambientales, tal como tiempo y ubicación. Pero es muy importante además tomar en cuenta información que puede ser aportada directamente por el usuario o por la interacción del mismo con las aplicaciones informáticas.

Tomamos como fundamento para nuestra propuesta la creación de un perfil para personalizar la experiencia del usuario, práctica usual en sistemas basados en la web y de recuperación de información. Este perfil puede ser alimentado explícitamente a través de datos recogidos de la interacción entre el usuario y el sistema; esta información contextual luego es integrada en el requerimiento de entrada y en la selección de contenidos, así como en la presentación de resultados [79].

Por otro lado, el trabajo presentado en Schmidt [132] nos sirve de guía para proponer el modelo de sistemas de Bases de Datos Difusas sensibles al

contexto, con una capa intermedia que gestione el contexto del usuario. El autor mencionado afirma que dicha capa intermedia es indispensable para abordar la complejidad en la gestión y la adquisición de información contextual de alto nivel, proporcionando aplicaciones que renueven la visión del contexto actual de los usuarios, materializando una parte y recuperando la otra dinámicamente desde otras fuentes.

De los antecedentes revisados, varias investigaciones de bases de datos se han enfocado en las preferencias contextuales. Cuando se define el término de preferencia se entiende que cada usuario en particular debe escoger las mismas y añadir hipotéticos contextos. Esto significa que estas preferencias pueden variar de acuerdo a la situación del usuario, sobrecargando al usuario en este tipo de definiciones que pueden ser incluso estériles si no se da la situación hipotética de ciertos contextos.

En nuestra propuesta nos enfocamos en gestionar el contexto actual y cómo afecta este contexto en la interacción entre el usuario y la bases de datos difusa, con el objetivo de mejorar la comunicación entre el usuario y el sistema, eliminando respuestas que pueden ser consideradas fuera de contexto o información ruido. Es por ello que consideramos nuestra propuesta abarca un espectro más general en sistemas sensibles al contexto, que incluye incluso a las presentadas hasta ahora para los sistemas de bases de datos.

Para evitar la sobrecarga en el usuario, en la definición de múltiples preferencias dependiendo de los hipotéticos contextos, algunas propuestas se han enfocado en inferir las preferencias, añadiendo complejidad a la interacción usuario bases de datos y herramientas que no son nativas de los SGBD, que añaden nuevas funcionalidades, personalizando y flexibilizando los sistemas pero con el riesgo de producir efectos indeseables, tales como: afectar el rendimiento de los SGBD, con el desarrollo de aplicaciones poco amigables e intrusivas al usuario ya que los usuarios necesitan conocer extensiones de los lenguajes de interacción con sintaxis que son añadidas al estándar.

En cambio, nosotros proponemos considerar el contexto como se ha hecho en otras áreas de computación (inteligencia ambiental, computación ubicua, sistemas de recuperación, interacción persona ordenador) para desarrollar aplicaciones sensibles al contexto, donde los resultados de las consultas puedan ser afectados de acuerdo al contexto en que son efectuadas, personalizando la experiencia del usuario sin sobrecargarlo de actividades que no sean de su interés, añadiendo nuevas funcionalidades que sean lo más transparentes posible, tratando de minimizar el impacto sobre el rendimiento del SGBD al utilizar herramientas estándares y operaciones ya definidas en los lenguajes de manipulación de datos.

Como un aspecto a ser tomado en cuenta en la propuesta, se destaca la importancia de contar con un registro o historial de las interacciones entre el

usuario y las aplicaciones, para de esta forma poder inferir cuáles son los objetivos, gustos o hábitos de los usuarios. Esta técnica es muy utilizada actualmente para sistemas de recomendación, recuperación de información y búsquedas interesantes para usuarios. Así, no dejamos de abordar el problema de las preferencias pero en el sentido de que un usuario o una aplicación que se conecte con el sistema pueden explícitamente especificar las mismas como contexto dependiendo del dominio de la aplicación.

Debido a que el usuario muchas veces debe especificar explícitamente el contexto, teniendo en cuenta que el lenguaje natural utilizado por los mismos puede ser ambiguo, subjetivo y por la naturaleza del proceso de comunicación es dependiente del contexto; entonces, es crucial la utilización de sistemas que nos permitan abordar el problema de representar información imperfecta tal como las Bases de Datos Difusas, cuya idoneidad para abordar este propósito ha sido demostrada en múltiples investigaciones y aplicaciones. Además, nuestra intención es ser lo menos intrusivos posible en el requerimiento de datos contextuales.

Queremos contribuir al desarrollo de sistemas inteligentes, proponiendo sistemas donde los usuarios puedan interactuar con bases de datos difusas, que permitan el almacenamiento de datos imperfectos y consultas flexibles, mejorando la experiencia del usuario al incorporar la información contextual produciendo sistemas más personalizados.

En el próximo capítulo presentaremos definiciones importantes de las herramientas que nos servirán para el posterior desarrollo de nuestro modelo, específicamente en el área de Bases de Datos Difusas y su implementación utilizando un Gestor de Bases de Datos Objeto-Relacional.

Capítulo 3

Bases de Datos Difusas

Los Sistemas Gestores de Bases de Datos (SGBD) se han extendido para almacenar nuevos tipos de datos tales como imágenes, sonidos, texto de gran tamaño, video interactivo, objetos multimedia y están preparados para responder consultas más complejas; se han creado almacenes que pueden consolidar una gran cantidad de datos provenientes de diversas fuentes (*Datawarehouse*) y se logran llevar a cabo análisis especializados mediante técnicas como procesos analíticos en línea (*OLAP*) o extracción de conocimiento utilizando minería de datos (*Datamining*).

También se están utilizando técnicas, que incluyen diversas áreas científicas, que permiten desarrollar proyectos para manipular gran cantidad de información tales como bancos de datos de genes humanos y sistemas de observación de señales del espacio; además se está trabajando con paradigmas como bases de datos distribuidas heterogéneas (donde los datos provienen de múltiples fuentes que pueden estar en diferentes esquemas y provenir de diversos SGBD) y la Web Semántica relacionando áreas de bases de datos (metadatos semánticos y ontológicos) con Inteligencia Artificial (agentes inteligentes).

Además, en los sistemas informáticos actuales se incluye el desarrollo de aplicaciones que utilizan Bases de Datos (tradicionales, de conocimiento o móviles) con herramientas tales como Modelos de Datos Semi-estructurados (como XML), Sistemas Gestores de Bases de Datos Objeto Relacionales (SGBDOR) y/o Sistemas distribuidos. Debido a paradigmas emergentes como Internet de las cosas, Web² y el amplio uso de las redes sociales, se producen un gran volumen de datos, por lo que a estos tiempos diversos autores los denominan la era de *Big data*; en esencia, la computación móvil y distribuida, los sistemas abiertos y las tecnologías basadas en la Web han cambiado radicalmente el modo de desarrollar aplicaciones de Bases de Datos.

Es por estas razones que cada vez con más frecuencia hay una gran cantidad de datos proporcionados por el mundo real que son imperfectos (vagos o flexibles), los cuales no son manipulados adecuadamente por los sistemas de bases de datos tradicionales. Aunado a ello la incorporación en nuestra vida cotidiana de dispositivos (sensores) para recolectar datos de video, sonido,

iluminación, movimiento, flujos de datos (*data streaming*), localización (proveniente de instrumentos GPS o imágenes) ha resultado en el incremento de la necesidad de gestionar estos tipo de datos.

En este capítulo se muestra un estudio de sistemas que van en esta dirección como son las de Bases de Datos Difusas (BDD). Se comienza con la revisión de los antecedentes, examinando los datos imperfectos y su representación, un breve marco teórico de lógica difusa, la implementación de BDD mediante los diversos modelos estudiados, para finalmente exponer una descripción más detallada del modelo de datos orientado a objeto para representar dominios difusos a utilizar en la presente investigación.

3.1. Introducción

Desde hace varios años, muchos investigadores han utilizado la Teoría de Conjuntos Difusos y Lógica Difusa para modelar e implementar la imperfección de los datos en los sistemas gestores de bases de datos como puede verse en la literatura [26] [27] [53] [113] [114]. La mayoría de estas propuestas están desarrolladas para ser implementadas en Sistemas Gestores de Bases de Datos Relacionales (SGBDR). Por otra parte, se han abordado diversas aproximaciones teóricas de modelos de datos conceptuales difusos, tal como lo muestra Ma y Yan [86], quienes efectúan una revisión de dichas propuestas.

Las bases de datos difusas se basan en la Teoría de Conjuntos Difusos desarrollada por Zadeh en su trabajo seminal de 1965, a partir del cual él mismo y muchos autores han investigado su uso y aplicación en diversas áreas, tales como la Inteligencia Artificial, la Ingeniería de Control, la Teoría de la Decisión, los Sistemas Expertos, la Lógica, la Ciencia de la Administración, la Investigación Operativa, la Robótica, entre otros; incluyendo específicamente el área de bases de datos. Desde su creación, la Teoría de Conjuntos Difusos ha avanzado en una variedad de direcciones y en muchas disciplinas.

Si bien ha tenido mucho éxito su aplicación en Ingeniería de Control de la industria manufacturera, existen también un gran número de desarrollos en el área de la Informática, ya que es uno de los pilares para el desarrollo de la Computación Flexible (*Soft Computing*) y Sistemas Inteligentes. Así el tema de Conjuntos Difusos, Lógica Difusa y su aplicación en sistemas informáticos y bases de datos es un área muy activa de investigación como lo demuestran trabajos recientes [5] [25] [45] [56] [59] [70] [106] [113] [117] [170].

También se han desarrollado aplicaciones en áreas tales como: Bases de datos Difusas Temporales [98] [57] [118], Gestión de datos semánticos [35], recuperación flexible de imágenes médicas [100], citas médicas [142], dietas para personas diabéticas [80], recursos humanos [79], toma de decisión en cultivo de

olivares [44], evaluación de profesores [3], selección de lugar para el planeo recreativo de aviones sin motor [129], clasificación de municipalidades de acuerdo a parámetros económicos y sociales [31], gestión inmobiliaria [13], así como en el turismo [81], por citar trabajos en áreas disímiles. Una recopilación reciente de investigaciones y aplicaciones significativas, la cual cubre estructuras difusas, sistemas, normas, operaciones, así como aplicaciones importantes en la toma de decisiones, la comunicación, la predicción y prevención ambiental; se encuentra en [89].

En la comunidad científica de bases de datos se han presentado muchas aplicaciones en diversas investigaciones, sin embargo el uso de las Bases de Datos Difusas no se ha extendido ampliamente en el mundo comercial. Entre algunas de las razones que han impedido que se popularice el uso de esta técnica de manera más amplia, se puede mencionar la necesidad de añadir capas lógicas de aplicaciones sobre los SGBDR (que por naturaleza son sistemas complejos) para gestionar los datos imperfectos lo que afecta el rendimiento computacional; además, el usuario debe aprender nuevas funciones y operadores para poder interactuar con las aplicaciones desarrolladas, por lo que son poco amigables y a veces intrusivas al usuario.

En este sentido nuestra propuesta pretende ser una contribución donde se utilizan herramientas estándares proporcionadas por sistemas gestores de bases de datos Objeto Relacionales (SGBDOR), con la intención de que sean lo menos intrusivos posibles, extendiendo las facilidades proporcionadas por los sistemas tradicionales, permitiendo desarrollar aplicaciones flexibles, amigables para el usuario y personalizadas.

Debido a su importancia para nuestra investigación, a continuación revisaremos en la próxima sección conceptos previos importantes para la representación de datos imperfectos en bases de datos y antecedentes que existen en la literatura que han enfrentado con éxito este problema.

3.2. Representación de datos Imperfectos

Investigadores como Vila et al. [153] consideran varios tipos de información imperfecta para ser incluida en una bases de datos, los cuales se presentan seguidamente.

- *Términos imprecisos dados como valores de atributos.* Ocurre cuando el dato es desconocido o el dominio del atributo es impreciso por naturaleza, como por ejemplo: Luis Pérez es *joven*, el diagnóstico de la enfermedad es *grave*, el estudiante José López está en sus años académicos *finales*.

- *Información incierta proveniente de la calidad pobre de algún dato.* Corresponde a aquellos casos en la cual se tiene perfectamente definidos los datos, pero no se tiene certeza de si es verdadero, tal como: creo con un 90 por ciento de probabilidad que Luis Pérez vive en Madrid, estoy casi seguro que el automóvil que produjo el accidente tiene la identificación GR-1234-D, es posible que el diagnóstico sea un resfriado.
- *Datos imprecisos e inciertos.* Este es el caso en el cual los dos tipos de vaguedad considerados anteriormente aparecen juntos. Ejemplos de ello son las expresiones: el diagnóstico de la enfermedad es *serio* en el 90% de los casos, es *casi seguro* que el estudiante Luis Pérez obtuvo una nota *alta* en Matemáticas.

Debido a la gran variedad de imperfecciones que pueden afectar a los datos, una forma de enfocarlo en bases de datos es tratar aquellos que tienen una apariencia flexible (*soft*), separándolos de los que tienen especificaciones precisas y completamente ciertas (información almacenada en bases de datos tradicionales) [16].

Es de hacer notar que la imperfección puede incluir otras deficiencias además de las anteriormente mencionadas, tal es el caso de datos erróneos o inconsistentes que pueden proporcionar sensores, provenientes de fuentes heterogéneas o resultados de técnicas como la consolidación; estos tipos de datos requieren de un tratamiento con un enfoque más específico, lo cual no es tema de esta investigación.

De acuerdo a Dubois [51] la borrosidad (*fuzziness*) no es un asunto de estética, ni un ingrediente para conformar construcciones formales áridas; *es una característica inevitable de la mayoría de sistemas humanísticos y deberá ser tratado como tal.* De ahí la adopción de este marco para el desarrollo de nuestra propuesta donde se quiere aumentar las facilidades de proporcionar un modelo de Bases de Datos Difusas haciendo énfasis en la perspectiva humana.

Otro punto importante afirmado por los autores Berzal et al. [17] es que muchas formas de imperfección involucran la noción de *gradualidad*, ya que en diversas situaciones los conceptos utilizados para describir una situación no corresponden a una realidad simple precisa, sino a una gradual. Por ejemplo, el término *joven* es gradual, una población dada no se puede dividir en dos conjuntos: los jóvenes y los que no lo son, es más, una persona en particular puede ser más o menos joven; algunas personas indudablemente son jóvenes, otras no lo son y hay un tercer grupo que está entre estos dos, donde los límites de los tres conjuntos no están precisamente definidos, de allí el término de borroso o difuso para estos tipos de conjuntos.

En su investigación Motro [104] indica que la imprecisión en los valores de los datos toma diferentes formas. Puede que el valor real del dato pertenezca a un conjunto específico de valores, por lo que es llamado dato *disyuntivo*. Si el conjunto al cual pertenece es el dominio completo entonces es *indisponible*, *desconocido* o *perdido*. Si cada uno de los valores candidatos está acompañado por un número describiendo la probabilidad que el valor sea verdadero (y la suma de todos ellos es uno) entonces se denomina *probabilístico*. Los datos probabilísticos incluyen, de acuerdo al autor antes mencionado, los anteriores tipos imprecisos además de los precisos; ocasionalmente, la información disponible en la ausencia de datos precisos es un término descriptivo, estos datos son denominados *difusos*.

La incertidumbre debida a la vaguedad es razonablemente modelada con la ayuda de herramientas como la Teoría de Conjuntos Difusos. En el caso particular de esta propuesta, debido al énfasis en el significado de términos imprecisos dados por el ser humano en un contexto específico (más que en la ambigüedad o inconsistencia de los datos), se va a utilizar un enfoque basado en la Teoría de Posibilidad propuesta por Zadeh [167], la cual define distribuciones de posibilidad basadas en conjuntos difusos como una restricción borrosa que actúa como limitación flexible sobre los valores que pueden asignarse a una variable.

La importancia de la teoría antes mencionada reside en el hecho de que la intrínseca borrosidad de los lenguajes naturales, como consecuencia lógica de las expresiones utilizadas es *posibilística* y no probabilística (es de carácter más subjetivo que estocástico), representando cuantiosa información sobre la cual se basan las decisiones del ser humano en un contexto particular. Además, una de las más importantes facetas del pensamiento humano es la habilidad de resumir información en *etiquetas* de conjuntos difusos lo cual provee una relación aproximada con los datos originales [164]; es por ello que se recurre a los postulados de Zadeh [165] [168] para modelar la percepción y los procesos de pensamiento humano, todo ello ligado al contexto de las personas.

Dado que la información proveniente del mundo real que debe ser almacenada en las bases de datos es frecuentemente difusa es necesario que los sistemas tradicionales enfrenten dos problemas importantes de investigación: la representación de información difusa en una bases de datos y la provisión de flexibilidad en consultas a bases de datos, mediante la inclusión de términos lingüísticos orientados al ser humano y obteniendo resultados con grados de pertenencia.

Es decir que no sólo se deben gestionar datos imperfectos, sino que la imprecisión también puede estar presente en solicitudes de recuperación de datos, los usuarios pueden formular consultas en términos imprecisos ya sea

intencionalmente o por necesidad, en este caso se denominan consultas flexibles o difusas.

Es posible desarrollar un modelo que suministre datos flexibles con el lenguaje de consulta estándar, con la semántica extendida para gestionar los datos imperfectos en las consultas. Por otro lado, se puede extender el lenguaje y los métodos de procesamiento de consultas para permitir consultas flexibles sobre bases de datos estándares. Nuestra propuesta es abordada con el primer enfoque ya que a través de modelar términos con semántica adaptable, utilizando la teoría de posibilidad y el lenguaje de consulta estándar, se formulan consultas que pueden contener términos difusos utilizando el contexto del usuario, para obtener resultados que incluyen la gradualidad.

El modelo relacional no tiene una manera efectiva de manipular datos imperfectos y como se ha demostrado cada vez es más importante proporcionar capacidades en los SGBD para gestionar tipos de datos complejos provenientes del mundo real. Al no poseer medios para modelar estos datos, en general, este modelo ignora los datos imperfectos y se enfoca principalmente en valores que son conocidos con certeza; la única forma de representar un valor desconocido o indefinido es mediante la utilización del símbolo especial "NULL", con el fin de dejar un campo de cierta tupla sin valor hasta que, una vez conocido, éste pueda ser ingresado. Esto constituye una forma muy genérica y poco flexible de tratar la gran cantidad de datos imperfectos que se requieren manipular hoy en día.

Muchas de las propuestas para gestionar datos imperfectos o flexibilizar consultas, a través de SGBDR, se centran en colocar capas lógicas de software por encima de éste, que haga las veces de interfaz con el usuario, traduciendo posteriormente a sentencias del modelo relacional (normalmente SQL) y luego devolver de nuevo los resultados a un ambiente flexible. Dado la complejidad de los SGBDR, esto supone costos adicionales que se manifiestan en un impacto notable en el rendimiento de estos sistemas. También hay propuestas de modelos de datos difusos que luego se implementan en capas lógicas, Nuestra propuesta va más con este enfoque utilizando las capacidades objeto-relacionales de ciertos gestores, con el fin de impactar lo menos posible en el rendimiento del mismo.

Como conclusión, es indudable que mediante la gestión de datos imperfectos (incluye el ingreso, actualización, eliminación, consulta y extracción de conocimiento) utilizando como base los SGBD se proporciona una mejor aproximación de las bases de datos implementadas en el mundo virtual con respecto al universo real de los usuarios. Nuestra propuesta está orientada a sistemas de bases de datos que puedan dar semántica adaptable a términos en el contexto del usuario, representando la imperfección (mediante la teoría de conjuntos difusos) de términos comúnmente utilizados en el lenguaje natural de las personas para contribuir en decisiones orientadas a la perspectiva humana.

3.3. Lógica Difusa y Representación de Datos Imperfectos

La lógica difusa tiene sus fundamentos en la Teoría de Conjuntos Difusos desarrollada por Zadeh [160]. En contraposición a los conjuntos clásicos donde los límites del mismo son precisos, los conjuntos difusos no tienen límites exactos; por lo tanto un elemento puede pertenecer parcialmente a un conjunto. Esto significa que la transición de pertenecer o no pertenecer a un conjunto es gradual, y esta transición continua es caracterizada por funciones de pertenencia; el grado de pertenencia a un conjunto difuso, denotado normalmente por la letra griega μ , le proporciona a los conjuntos difusos la flexibilidad de modelar expresiones lingüísticas comúnmente usadas que reflejan mejor la naturaleza de los pensamientos humanos los cuales tienden a ser abstractos e imprecisos.

De acuerdo a Zadeh [164] el término de *lógica difusa*, usado por primera vez en 1974, tiene dos alcances diferentes: en un sentido limitado, es visto como una extensión de la lógica multivaluada con el propósito de servir para el razonamiento aproximado; en un sentido más amplio, es un sinónimo de la teoría de conjuntos difusos.

Es importante reconocer que el término *lógica difusa* es usado predominantemente en su sentido más amplio, así cualquier campo X puede ser generalizado (*fuzzified*), reemplazando el concepto de un conjunto preciso en X por un conjunto difuso. Es por ello que agrupa además de la teoría de conjuntos difusos, reglas si-entonces [45], aritmética difusa [136], cuantificadores [46], entre otros. Su éxito se debe principalmente a la posibilidad de resolver problemas de una gran complejidad y poco definidos que, mediante métodos tradicionales, son difíciles de solucionar.

En la lógica booleana sólo existen dos valores para el grado de pertenencia a un conjunto: 0 para denotar aquellos elementos que no pertenecen a un conjunto definido y 1 para el caso contrario; así, un elemento no puede pertenecer en forma simultánea a un conjunto y a su complemento. Esta característica permite tomar estos casos como dos de las posibilidades de pertenencia de un elemento a un conjunto difuso. Sin embargo, en estos últimos pueden existir elementos que pertenecen a un conjunto y simultáneamente a su complemento *en algún grado*. Por esta razón la lógica difusa se aplica especialmente a términos lingüísticos que no poseen límites claramente definidos tales como estatura (alto, medio, bajo, enano), peso (gordo, adecuado, delgado), edad (muy joven, joven, maduro, viejo, muy viejo) o proximidad (lejano, medio, cercano).

A continuación, en la próxima sub-sección, se muestran conceptos básicos, ejemplos, terminología importante de Lógica Difusa y su aplicación en Bases de Datos que consideramos necesarios para nuestro estudio. Para el lector interesado existen diversas publicaciones para profundizar en conceptos de Lógica Difusa,

Teoría de Conjuntos Difusos, variables lingüísticas, etiquetas lingüísticas, Teoría de la Posibilidad; Computación con Palabras y Teoría Computacional de Percepciones: además de los trabajos de Zadeh, se pueden revisar estudios de Seising [134], y más recientemente de McNeil y Ellen [96], Terano, Asai y Sugeno [142].

3.1.1. Conceptos Básicos

En primer lugar se definirá el conjunto difuso y la función de pertenencia para posteriormente presentar algunos ejemplos de esta última, la cual puede tener diversas formas, dependiendo de la interpretación y el cálculo de la misma; además, a través de otros ejemplos, se presentan los conceptos de variables y etiquetas lingüísticas.

Definición 1. *Conjunto difuso y función de pertenencia*

Sea X una colección de objetos cuyos elementos se denominan genéricamente x ; entonces un *conjunto difuso* A en X es definido como un conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\},$$

Donde $\mu_A(x)$ es llamada la función de pertenencia (FP) para el conjunto difuso A . La FP asocia cada elemento de X a un grado de pertenencia entre 0 y 1.

En el ejemplo 1 se presenta la definición de un conjunto difuso y luego se representa su FP.

Ejemplo 1. *Personas de estatura “alta”*

Se puede afirmar que personas de estatura menor a 1,60 metros no son altas, mientras que las mayores de 1,90 son altas. Gráficamente se puede ver la interpretación de esta definición tal como se muestra en la figura 2.

Usualmente el dominio X es referido como el *universo de discurso* o *universo*, el cual puede ser constituido de objetos discretos (ordenados o desordenados) o espacio continuo; seguidamente se presentan ejemplos de cada uno de ellos.

Ejemplo 2. *Conjuntos difusos con un dominio discreto no ordenado.*

Sea $X = \{\text{Madrid, Barcelona, Valencia, Granada}\}$ el conjunto de ciudades que podrían escogerse para vivir. Se define el conjunto difuso A como la *ciudad deseable para vivir* descrito como sigue:

$$A = \{\text{Madrid/ 0.7, Barcelona/ 0.8, Valencia/ 0.9, Granada/ 0.85}\}$$

El universo de discurso X es discreto, contiene objetos sin orden (ciudades) y la FP definida es muy subjetiva; cualquier persona que no sea este autor puede

definir grados de pertenencia distintos pero legítimos que reflejen *su preferencia*.

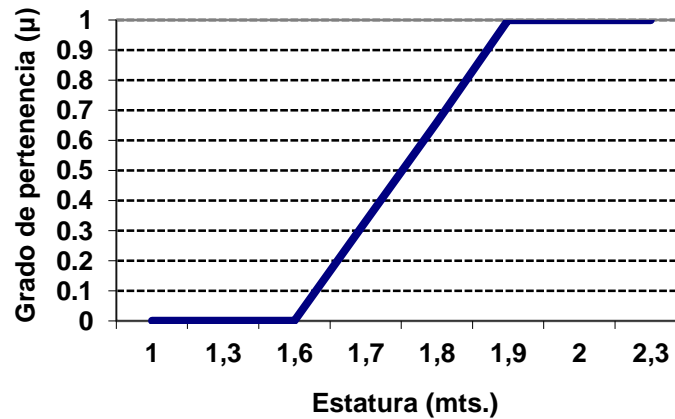


Figura 2. Función de Pertenencia de una persona de estatura “alta”

Ejemplo 3. Conjuntos difusos con un dominio discreto ordenado.

Sea $X = \{0, 1, 2, 3, 4, 5\}$ el conjunto de número de hijos que una familia escoge tener y el conjunto difuso A como *el número razonable de hijos en una familia*, descrito como sigue:

$$A = \{(0; 0.1), (1; 0.7), (2; 1), (3; 0.7), (4; 0.3), (5; 0.1)\}$$

En este caso se tiene un universo discreto X , la FP para el conjunto difuso A se muestra gráficamente en la Figura 3, donde los grados de pertenencia son, de nuevo, subjetivos.

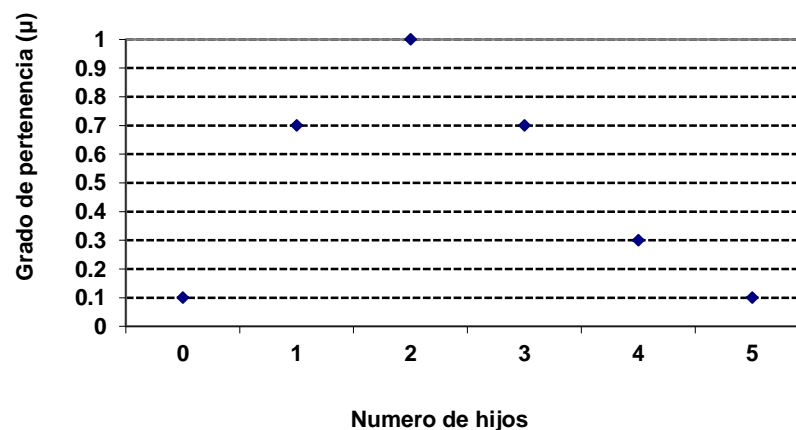


Figura 3. Interpretación de número razonable de hijos en una familia

Ejemplo 4. Conjuntos difusos con un dominio continuo.

Sea X símbolo de contenido \mathbb{R}^+ el conjunto de posibles edades de seres humanos. Entonces el conjunto difuso A definido como *alrededor de los 50 años*,

puede expresarse como sigue:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

$$\text{donde, } \mu_A(x) = \frac{1}{1 + \left(\frac{x-50}{10}\right)^4}$$

Esta interpretación es mostrada en la Figura 4. Nótese que en este caso se dibuja una curva dando la noción de gradualidad, pero en la práctica, debido a la facilidad de su manipulación matemática y porque no pierde el significado de lo que se desea simbolizar, en la comunidad de investigadores de bases de datos se acepta su representación gráfica a través de varias rectas, constituyendo una forma trapezoidal o triangular.

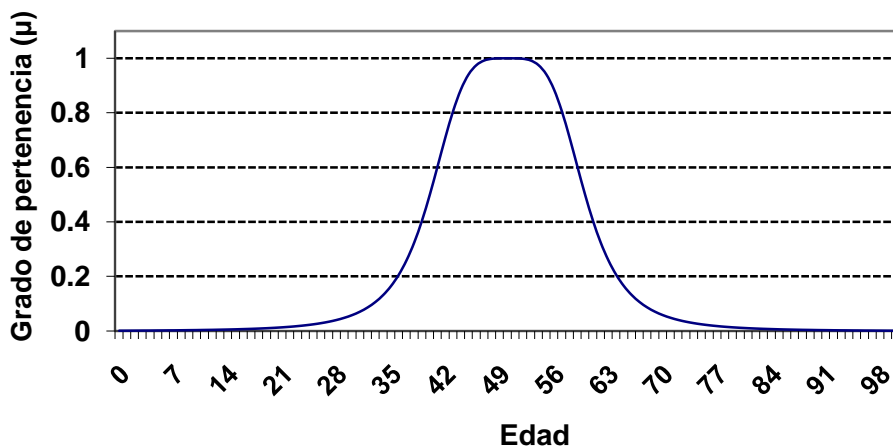


Figura 4. Interpretación de “alrededor de 50 años de edad”

De lo expuesto hasta ahora se puede notar que la definición de conjuntos difusos depende de dos aspectos fundamentales: la identificación de un universo de discurso deseable y la especificación de una adecuada función de pertenencia; ésta última, en muchas ocasiones, es subjetiva, debido a que la expresión de conceptos abstractos depende de la percepción individual o del *contexto del usuario*; además tiene poco que ver con la aleatoriedad; estas dos características diferencian a los conjuntos difusos con la teoría de probabilidades, ya que ésta última tiene como finalidad el tratamiento objetivo de fenómenos aleatorios.

En la práctica, cuando el universo de discurso X es un espacio continuo, normalmente se divide en varios conjuntos difusos cuyas FPs cubren X en una forma más o menos uniforme. Estos conjuntos difusos usualmente llevan nombres que conforman adjetivos que se utilizan diariamente (grande, pequeño o mediano) los cuales son llamados *valores o etiquetas lingüísticas*, por lo que el universo de discurso X es llamado frecuentemente la *variable lingüística* [160]. Estos

conceptos permiten modelar el pensamiento humano, además de servir para generalizar información y expresarla en términos de conjuntos difusos. Debido a la importancia para nuestra investigación a continuación se presentan la definición.

Definición 2. *Variable lingüística.*

Es una variable cuyos valores son palabras u oraciones en un lenguaje natural o artificial. En términos más formales, es caracterizada por una quintupla $(N, T(N), X, G, M)$; en el cual N es el nombre de la variable, $T(N)$ es el conjunto de términos o valores lingüísticos, X es un universo de discurso, G es una regla sintáctica que genera los términos en $T(N)$, y M es una regla semántica que asocia a cada valor lingüístico L su significado $M(L)$.

Ejemplo 5. *Variable lingüística y valores lingüísticos.*

Suponga que el universo de discurso X es la edad de una persona, se define conjuntos difusos para *joven*, *maduro* y *viejo*; caracterizados por las FPs $\mu_{joven}(x)$, $\mu_{maduro}(x)$ y $\mu_{viejo}(x)$ respectivamente. Así como una variable puede asumir diversos valores, una variable lingüística tal como *edad* puede tener diferentes valores lingüísticos, tales como *joven*, *maduro* o *viejo*. FPs típicas para estos valores lingüísticos son mostradas en la Figura 5, donde el universo de discurso X es totalmente cubierto por la FPs y la transición de una FP a otra es de forma suave y gradual.

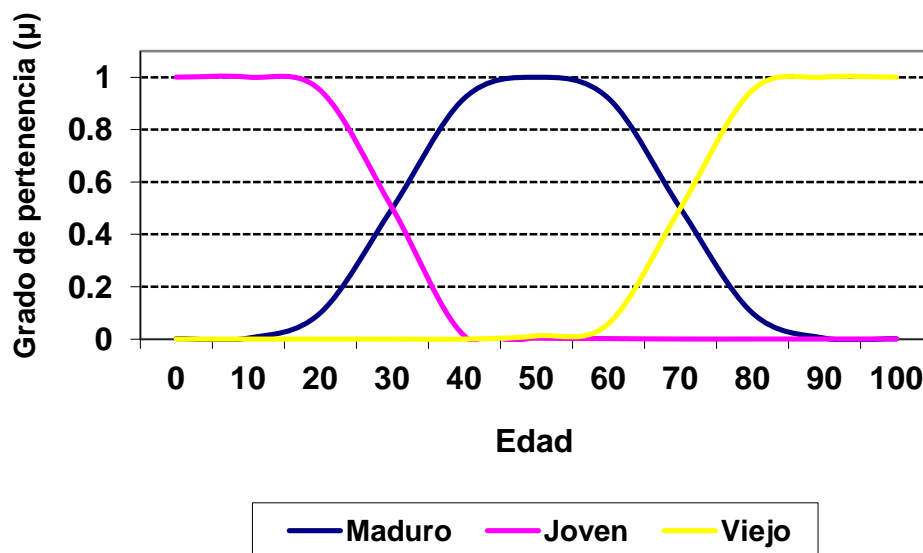


Figura 5. FPs de valores lingüísticos “maduro”, “joven” y “viejo”

El concepto de una variable lingüística juega un rol esencial en las aplicaciones de lógica difusa debido a que asemeja la forma en la cual los humanos perciben, razonan y se comunican. El uso de palabras puede ser visto

como una forma de síntesis de datos numéricos que explota la tolerancia a la imprecisión para cumplir con los postulados de la computación flexible [168].

En este sentido, de acuerdo a Berzal et al. [16], algunos conceptos de *computación con palabras*, término previamente introducido por Zadeh [165], han sido utilizados para extender modelos de bases de datos (relacional, orientado a objetos y objeto relacional) a fin de permitir la representación de datos imperfectos. Es por ello que nuestra propuesta sea una contribución en este sentido utilizando el modelo objeto relacional para flexibilizar el uso de términos descriptivos dependiendo del contexto del usuario.

Es así como Zadeh [168] explica que una variable lingüística es interpretada como una *etiqueta* de un conjunto difuso que es caracterizado por una función de pertenencia, luego si x es una edad numérica como 53, entonces $\mu_{maduro}(53)$ es el grado de pertenencia a *maduro*. Lo importante es que esto constituye una medida subjetiva, ya que la interpretación de $\mu_{maduro}(x)$ es el grado que establece a x la *percepción* que para el concepto maduro tiene una persona en un *contexto dado*.

De allí nuestro interés de establecer múltiples contextos personalizados para los usuarios, lo que va a permitir establecer semánticas adaptables para los conceptos manipulados en un ámbito de incertidumbre modelado por Bases de Datos Difusas utilizando un SGBDOR.

De acuerdo a Zadeh [164] la mayoría de las aplicaciones que utilizan lógica difusa emplean un patrón simple de valores lingüísticos como el mostrado en la figura 6. Específicamente asumen funciones de pertenencia tipo triangular o trapezoidal y el número de valores o etiquetas lingüísticas está usualmente en el rango de tres a siete.

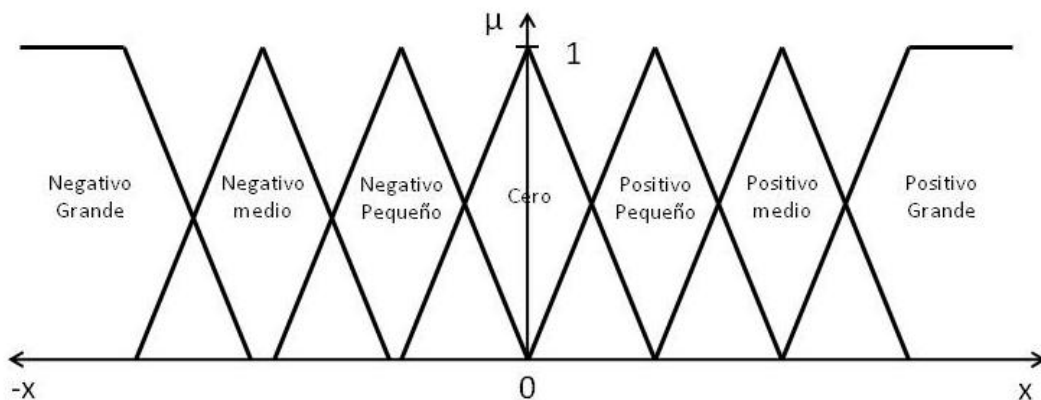


Figura 6. Valores lingüísticos triangulares usuales, adaptado de Zadeh [164]

Es importante destacar que estas funciones de pertenencia tienen que cumplir con las características de normalidad y convexidad (números difusos). Aunque Pedrycz [110] justifica en su artículo la utilización de funciones

triangulares (y trapezoidales) en muchas aplicaciones de controladores difusos, modelos difusos y esquemas de clasificación; es indudable que para nuestra propuesta este tipo de funciones no tiene que seguir obligatoriamente este patrón, porque cada usuario será libre de definir los conceptos con etiquetas personalizadas de acuerdo a su contexto.

De igual forma, estudios de antropología y el pensamiento humano indican que es poco probable la utilización de más de siete etiquetas; normalmente están en este rango o menos, como por ejemplo: *enorme, muy grande, grande, normal, pequeño, muy pequeño o ínfimo*. La idea final es que el modelo a seguir, por aplicaciones de *soft computing* como la propuesta en esta investigación, sea la mente humana de acuerdo a los postulados de Zadeh [165].

Las FP tipo trapezoidales (cerradas o abiertas) junto con las triangulares ya descritas constituyen la base de la mayoría de las utilizadas para representar variables lingüísticas modeladas por el ser humano, aunque también hay FP que pueden ser definidas por funciones tal como se hizo en el ejemplo 4 para *alrededor de los 50 años de edad*.

Zadeh propone una metodología en la cual las palabras son usadas en lugar de los números para computación y razonamiento, donde la lógica difusa juega un papel primordial en computación con palabras (CW) y viceversa [165]. CW es una necesidad cuando la información disponible es imprecisa; además, la tolerancia a la imprecisión puede ser explotada para lograr soluciones tratables, robustas, de bajo costo y relacionadas con la realidad. Estos postulados son base para la Teoría Computacional de Percepciones (TCP) propuesta por Zadeh [168], que sirvió de fundamento para el desarrollo de los módulos de almacenamiento y recuperación de datos propuestos en esta investigación. De allí el nuestro interés en el estudio de estas teorías.

3.1.2. La Lógica Difusa en Representación de la Información

En una revisión sobre el uso de conjuntos difusos en las áreas de bases de datos y recuperación de información elaborada en el año 2005 por los investigadores Bosc, Kraft y Petry [27], se estudian los aspectos de esta área que parece tener mayores potencialidades para su desarrollo futuro, incluyendo diseño de bases de datos, preferencias para consultas flexibles, dependencias funcionales difusas y redundancia.

Hay dos aspectos de conjuntos difusos de particular interés para las bases de datos: 1) La noción de coincidencia graduada, no relacionada a la incertidumbre, sino a valores de verdad los cuales tiene que ver con gradualidad, y 2) La idea de preferencias sobre candidatos para una variable cuyos valores

precisos es desconocido (teoría de posibilidad de Zadeh). En este caso, la incertidumbre es un asunto central debido a que cuando el valor tomado por esta variable es cuestionado, la respuesta no puede darse con seguridad.

Los aspectos mencionados han determinado un amplio rango de tópicos en el área de bases de datos, tales como: 1) Integrar imprecisión en los modelos de datos (primero en modelos relacionales, luego en orientados a objeto), dando lugar a lo que se denominan bases de datos difusas, 2) Implementar bases de datos difusas, 3) Consultar bases de datos que contienen información imprecisa, 4) Definir y procesar consultas flexibles dirigidas a bases de datos tradicionales, y 5) Extender propiedades tales como dependencias funcionales o multivaluadas, y más generalmente expresar propiedades para propósitos de extracción de conocimiento [27].

De acuerdo a Vila et al. [153] la lógica difusa y teoría de conjunto difusos propuesta por Zadeh [160], constituyen un buen marco para representar diferentes tipos de imprecisión e incertidumbre encontrado en los datos reales. Estos autores reflexionan acerca de diferentes tipos de datos imperfectos que pueden aparecer en un problema de representación del mundo real, además del rol jugado por las teorías antes mencionadas en el contexto estudiado. Los conjuntos difusos han sido utilizados en áreas relacionadas tales como minería de bases de datos, bases de datos espaciales, sistemas geográficos de información y de recuperación de información, entre otros.

A partir de ahora nos centraremos en el estudio de la incorporación de la lógica difusa a distintos modelos de Bases de Datos.

3.1.3. Bases de Datos Relacionales Difusas

Desde hace años, diferentes autores han trabajado sobre bases de datos relacionales difusas. Una de las primeras propuestas en esta área es la de Tahani [140], limitándose sólo a consultas difusas sobre bases de datos tradicionales. Luego investigadores como Buckles y Petry [29] extendieron las operaciones primitivas del modelo relacional de Codd [39], asociando grados de pertenencia a las tuplas, distribuciones de posibilidades a valores de atributos y formulando un marco teórico donde las tuplas se comparan por relaciones de similitud; este enfoque lo denominaron modelo relacional de distribución de posibilidad.

Destaca también Umano [147], quien propone un Sistema de Bases de Datos difusas denominado Freedom-0 que permite representar datos difusos a través de distribuciones de posibilidad y números difusos utilizando un lenguaje de programación embebido en Fortran. Posteriormente Zemankova-Leech y Kandel [174] desarrollan una bases de datos relacional difusa (BDRD), la cual

puede manipular tanto los conjuntos difusos como distribuciones de posibilidad, efectuando extensiones a las estructuras de datos para que el componente de una tupla no esté restringido a valores atómicos.

Prade y Testamale [119], Testamale [143] extienden las bases de datos relacionales para permitir valores difusos en los atributos, utilizando distribuciones de posibilidad; así manipulan relaciones ordinarias en productos cartesianos de conjuntos de subconjuntos difusos en lugar de relaciones difusas, por lo que pueden tomarse en cuenta también consultas vagas representadas por distribuciones de posibilidad. En un enfoque teórico, un modelo de datos de relación difusa sencillo es presentado por Raju y Majumdar [122] y una aplicación del modelo de Buckles y Petry es realizada por Sheno, Melton y Fan [135].

El modelo básico de una bases de datos relacional difusa (BDRD) en su forma más simple consiste en añadir un grado (en el intervalo $[0,1]$) a cada tupla, permitiendo mantener los datos de la bases de datos en su forma original; la semántica que se le asigna a dicho grado será la que determine su utilidad en los procesos de consulta.

Como en relaciones clásicas, una relación difusa Rd es representada como una tabla con una columna adicional para $\mu_{Rd}(t_i)$ denotando el grado de pertenencia de la tupla t en Rd . Más aún, esta tabla contiene sólo aquellas tuplas para las cuales $\mu_{Rd}(t) > 0$; es decir, para cada tupla no presente en la tabla se supone que $\mu_{Rd}(t) = 0$. Esta observación puede ser considerada como una versión difusa de la asunción de mundo cerrado.

Una nueva definición de bases de datos relacionales utilizando conceptos lógicos es presentada por Vila et al. [153]. Con este enfoque es posible considerar diversos tipos de borrosidad (*fuzziness*) para los atributos de las bases de datos; también establecen un lenguaje de consulta basado en cálculo de dominio para el modelo, permitiendo formular varios tipos de consultas con diferentes niveles de carencia de precisión.

En ese mismo año Medina, Pons y Vila [99] proponen un modelo generalizado para BDRD (GEFRED), definiendo nuevos atributos que extienden las bases de datos convencionales a difusas, llegando a instancias de definición de conceptos, constituyendo además una síntesis de los modelos en esta área hasta la fecha publicados. El objetivo era resolver los problemas de representación y manipulación de información difusa tomado en cuenta su naturaleza específica, permitiendo al usuario escoger el operador comparación (de necesidad o posibilidad) y la medida de compatibilidad difusa a ser usada en una consulta.

A partir de GEFRED se propone el lenguaje Fuzzy SQL (FSQL) extendiendo al SQL clásico para expresar valores difusos, condiciones difusas y atributos difusos, entre otros. Para interpretar este lenguaje se requiere de un servidor difuso (FSQL) y una meta base de conocimiento difuso (FMB),

traduciendo las sentencias difusas al lenguaje soportado por las bases de datos estándar. Una implementación muy completa utilizando el SGBD Oracle se puede encontrar en Galindo [54], posteriormente mediante PostgreSQL en Urrutia, Galindo y Sepúlveda [148]. Además el FSQL es reconocido en el área de investigación como uno de los lenguajes más completos para la interacción con un sistema de bases de datos relacional difuso.

Este modelo ha sido la base para realizar diversas aplicaciones prácticas de Bases de Datos Difusas, tal como lo muestran Touzi y Hassine [146] o Rodrigues, Cruz y Cavalcante [123]; además de ser utilizado en arquitecturas de integración mediante ontologías en Blanco, Martínez-Cruz y Vila [20]; también en propuestas de minería de datos difusas mostradas en Carrasco, Vila y Araque [36] o Blanco, Martínez-Cruz, Serrano y Vila [22].

También es de destacar los trabajos hechos sobre consultas flexibles, sin implementar atributos difusos en las bases de datos, es decir, manteniendo los datos tradicionales de la bases de datos relacional e incorporando imprecisión sólo a nivel de consultas. Una de las investigaciones más destacadas es la propuesta de SQLf realizada por Bosc y Pivert [28].

SQLf es un lenguaje de consultas difusas diseñado en una capa lógica sobre los SGBDR, consintiendo el uso de una condición difusa en cualquier lugar donde se permite una condición booleana clásica. Tales condiciones involucran etiquetas lingüísticas con una semántica definida por el usuario que expresa las preferencias del mismo. Se han realizado aplicaciones bajo este marco utilizando diversos SGBDR como Oracle, DB2, Firebird, PostgreSQL, MySQL denominado SQLfi reportado por Goncalves y Tineo [60], así como una implementación de mayor rendimiento realizada en la capa física a través de un acoplamiento fuerte con PostgreSQL mostrada en Cadenas, Aguilera y Tineo [32].

Hay aportes como el de Zhao, Ma y Yan [175] que proponen una extensión del modelo relacional para representar y tratar con información difusa por medio de conjuntos vagos. Presentan una estructura revisada y extendida del álgebra relacional; demuestran que ésta es cerrada y reducible al álgebra relacional difusa, además de la convencional.

Todos los antecedentes estudiados se consideran de importancia en la evolución de las bases de datos difusas, tomando en cuenta la gran cantidad de modelos teóricos, arquitecturas e implementaciones realizados sobre BDRD. Como se ha mencionado anteriormente esta revisión documental no cubre toda la gama de propuestas existentes; sino las que consideramos, a nuestro juicio, más relevantes.

En la práctica no se ha extendido el uso de BDRD en forma comercial, empresarial o gubernamental; se cree que la arquitectura de capas lógicas que se deben diseñar sobre los SGBDR (lo que influye indudablemente en la eficiencia

del mismo), además de que el usuario debe aprender nuevas funciones y operadores, han impedido una mayor popularización de estas técnicas. Sin duda, detrás está también el difícil encaje de la subjetividad del usuario en los diseños de dominios establecidos por el diseñador. En este sentido, este trabajo de investigación supone un avance en la dirección adecuada.

3.1.4. Bases de Datos Orientadas a Objeto Difusas

A partir de los años 80 se desarrolla un intenso trabajo de investigación en modelos orientados objetos y su contraparte OODBMS, pensando que este paradigma iba a desplazar al modelo relacional en los siguientes años, así como ocurrió anteriormente con el modelo relacional respecto a sus antecesores (jerárquico y de red).

Es así como se desarrollan una gran cantidad de trabajos en las siguientes tres décadas que se pueden encontrar en la literatura de modelos orientados a objetos difusos para representar y gestionar estos tipos de objetos, constituyéndose de hecho en un área de investigación propia dentro de bases de datos.

Lo fundamental en este modelo son los conceptos provenientes de la programación orientada a objetos; así son importantes la encapsulación del estado (estructura) y la conducta (comportamiento) de un ítem dado del mundo real. La estructura y el comportamiento de objetos similares son agrupados en clases (que pueden reusarse), las cuales son organizadas en jerarquías, donde las subclases heredan las definiciones (estructurales y de comportamiento) de las superclases. Los distintos OODBM difusos han tratado la borrosidad (*fuzziness*) en diferentes niveles: dominios, instancia de relaciones, relaciones de herencia, estructura y comportamiento.

En las próximas sub-secciones se van a describir sólo, a nuestro juicio, los más destacados.

3.1.4.1 Modelos Conceptuales

La primera propuesta la presentan Zvieli y Chen [177] en el área sólo de modelado, extendiendo el modelo entidad-relación (ER) para permitir atributos difusos en entidades y relaciones a tres niveles:

- Conjuntos de entidades, relaciones y atributos pueden ser difusos, es decir, tener un grado de pertenencia en este nivel.

- El segundo nivel es relacionado a las ocurrencias difusas de entidades y relaciones.
- El último, es la representación de valores difusos de los atributos de especial interés en entidades y relaciones.

Asimismo Ruspini [127] describe los diferentes tipos de información vaga que un modelo de datos puede incluir, proponiendo su representación en el modelo de dato ER. Además considera escalas probabilísticas y posibilísticas para expresar medidas sobre la veracidad de los datos. Por otra parte Vanderberghe y de Caluwe [152] también presenta una extensión del E/R con nociones más elaboradas de conjunto difuso de entidades, subclases difusas y categorías difusas.

Basándose en jerarquías de clases, George et al. [58] modelan la incertidumbre utilizando lógica difusa para generalizar igualdad a similitud permitiendo que la incertidumbre en los datos sea representada por incertidumbre en la clasificación.

Yazici y George [158] presentan dos modelos conceptuales de información compleja e incierta denominados ExIFO u ExIFO₂ extendiendo sus propuestas anteriores. Utilizan valores de atributos difusos, incompletos y nulos; para cada uno de ellos hay una definición formal y una representación gráfica, presentando una serie de reglas de verificación para chequear la correctitud del esquema conceptual preservando algoritmos de transformación para modelos lógicos.

En su trabajo Vila et al. [154] plantean un modelo semántico de datos para la representación de información imprecisa en bases de datos orientada a objeto. La representación se basa en el modelo semántico de bases de datos formal (IFO) propuesto por Abiteboul y Hull [1] y en el problema de la incertidumbre presentado por Ruspini [127] e incorporan nuevos elementos gráficos para representar diferentes niveles de vaguedad. Afirman que es posible tratar con información difusa por medio de herramientas de software clásico, implementando un Sistema de Bases de Datos difuso a través de un OODBMS clásico, con el único requisito de agrupar y agregar conceptos, además de una vista funcional de los atributos.

Una extensión del modelo E/R para representar objetos imperfectos además de complejos del mundo real a nivel conceptual lo presentan Ma et al [87]. En particular, proporcionan un método formal para hacer concordar un modelo entidad-relación difuso extendido (FEER) con un esquema de bases de datos orientado a objetos difuso, justificando que ambos tipos de modelo juegan un papel crucial en el diseño de bases de datos relacionales y bases de datos orientados a objetos, respectivamente. Posteriormente Ma [85] presenta una recopilación de artículos que muestran los avances en Bases de datos Orientadas a Objeto Difusas.

Un artículo de revisión de modelos de datos conceptuales difusos propuestos en la literatura lo publican Ma y Yan [86]; se discuten principalmente los modelos de datos difusos ER/FER, IFO y UML; además examinan las aplicaciones de los modelos de datos conceptuales difusos. Ellos afirman que la teoría de conjuntos difusos ha sido ampliamente aplicada para extender diversos modelos de bases de datos, lo que ha dado lugar a numerosas contribuciones, principalmente con el popular modelo relacional o con alguna forma vinculada al mismo. Además justifican el uso de modelos de datos semánticos (conceptuales) difusos debido a que múltiples investigadores se han concentrado recientemente en él para satisfacer la necesidad de modelar objetos complejos con imprecisión e incertidumbre.

3.1.4.2 Modelos Orientados a Objeto

En uno de los modelos propuestos más completos encontrado en la literatura denominado UFO, los autores Van Gyseghem y de Caluwe [150] describen un esfuerzo de procesar información imperfecta en aplicaciones de bases de datos en una forma flexible, extendiendo un modelo orientado a objetos para definir conceptos flexibles, con capacidades de manejar amplias formas de imperfección; también es mostrada una implementación práctica a través de una interfaz a un modelo de bases de datos relacional extendido.

Posteriormente Rossazza, Dubois y Prade [125] presentan una representación centrada en el objeto, donde puede especificarse un rango de valores permitidos y un rango de valores típicos que pueden ser difusos; varios tipos de relaciones de inclusión (graduales) pueden definirse entre clases. También discute mecanismos de herencia así como otros tipos de razonamiento como clasificación.

También con diversas investigaciones en el área Bordogna, Lucarella y Passi [24] proponen un modelo para gestionar datos precisos y difusos (FOOD), basándose en grafos para representar la imperfección a nivel de atributos y relaciones de los objetos. Para formular consultas imprecisas y recuperar objetos exactos o imprecisos con un cierto grado, se gestiona la incertidumbre en dos niveles: en los datos y en el conocimiento de los mismos; estas capacidades son requisitos de muchas aplicaciones que tratan con datos de diferente naturaleza y con interrelaciones complejas. El modelo se basa en un paradigma visual el cual apoya la representación de datos semánticos y la directa visualización de la información; las instancias son representadas como grafos etiquetados dirigidos en la cual la información difusa e incierta tiene su propia representación.

Blanco et al., en su artículo [19], muestran cómo las diferentes fuentes de vaguedad pueden ser gestionadas en un sistema de bases de datos orientado a

objetos (OODB) tradicional, describen nuevas estructuras que deben ser incorporadas y utilizan el lenguaje de modelado unificado (UML) para la representación conceptual de dichas estructuras, debido a que claramente puede traducirse directamente a un modelo orientado a objeto.

Para modelar imperfecciones en el dato basado en el concepto de conjuntos difusos nivel-2, presentan su propuesta De Tré y De Caluwe [43]. En su trabajo se enfocan en las definiciones semánticas de los aspectos estructurales y de comportamiento de los datos, demostrando que el concepto de *tipo* puede ser generalizado por conjuntos difusos tipo-2. Debido a que los tipos son habitualmente reconocidos como los bloques básicos de construcción de modelos orientados a objetos, los tipos generalizados pueden ser usados como nociones básicas de modelos de bases de datos orientados a objetos difusos. También ilustran y discuten como el modelo de dato estándar de objetos para persistencia, desarrollado por el grupo de gestión de objetos de bases de datos (ODMG), puede ser generalizado para manipular datos del mundo real en una forma más ventajosa.

Un conjunto de operadores útiles para comparar objetos en un ambiente difuso lo presentan Marín et al. [91]. Aquí se introduce la noción de grado de semejanza generalizado para comparar objetos difusos complejos dentro de una clase, considerando recursividad en el caso de que un atributo de un objeto difuso sea a su vez un objeto difuso. Esta propuesta es considerada de sumo interés porque sienta las bases para problemas de comparación relacionados con la gestión de objetos difusos en OODB.

Berzal et al. [17] indican que el modelo orientados a objeto es más poderoso que el relacional para la representación de esquemas cuando existen relaciones complejas como las del mundo real, es por ello que surgieron muchas investigaciones que proponían mejoras de la orientación a objeto con conceptos difusos denominándolos modelos de bases de datos orientados a objeto difusos (FOODBM).

Los autores anteriormente mencionados introducen una arquitectura que puede ser usada para desarrollar un sistema capaz de almacenar información difusa en sistemas orientados a objeto convencionales. Este marco logra ser utilizado transparentemente a través de una interfaz a fin de proporcionar a los usuarios capacidades de gestión de datos difusos orientados a objeto.

Si se desea una profundización, con análisis crítico, ventajas y desventajas de los modelos orientados a objetos difusos más importantes como los mencionados anteriormente se puede consultar la disertación doctoral de Marín [90]. Además, el modelo que se propone en dicha tesis, basado a su vez en lo formulado por Vila et al. [155] que posteriormente extienden en [41], es el que se va a utilizar en la propuesta de esta investigación. Por la relevancia del estudio del mismo será estudiado en más detalle en una sección posterior de este capítulo.

Después de la revisión documental efectuada, es clara la ventaja de representación del modelo orientado a objeto sobre el relacional, cuando se trate de aplicaciones que involucren complejidad, dinamismo y además requieran datos imperfectos. Se ha identificado los tipos de vaguedad, presentes en sistemas de bases de datos orientados a objeto difusos, distinguiéndose en diversos niveles: valores de los atributos, entidades o clases, las interrelaciones entre clases (incluyendo herencia) y definiciones de tipos flexibles.

Es de hacer notar el énfasis que se hizo en la revisión de los modelos orientados a objetos y no en los OODBMS, ya que el interés de la propuesta es implementarlo en un SGBDOR, donde el modelado es orientado a objetos y aplican los conceptos presentados en las secciones anteriores. Por otra parte cabe resaltar que los OODBMS no tuvieron el éxito esperado a nivel de organizaciones (comercial, empresarial o gubernamental) siendo utilizados normalmente en entornos científicos y en ciertas universidades.

3.1.4.3 Modelos Objeto Relacionales y Otras Extensiones

Como ya se ha mencionado a lo largo de toda la investigación, la información imprecisa e incierta existe en muchas aplicaciones de la vida real y por esta razón el modelo de datos difuso ha sido extensivamente investigado para ser utilizado teniendo como fundamento diversos modelos de datos tradicionales.

Por ello, además de los modelos estudiados en las secciones anteriores, existen una serie de propuestas de modelos de bases de datos difusos utilizando un Modelo Objeto-Relacional, además de otras extensiones basadas en ontologías y en XML, para ser implementados ya sea en SGBDR o en OODBMS. En esta sección empezaremos con ejemplos puntuales de otras extensiones para enfocarnos luego en el de mayor interés para este estudio que son los Objeto-Relacionales.

Una ontología que permite a la estructura difusa de un sistema de bases de datos difuso (FDBMS) estar disponible en la Web Semántica es presentada por Blanco et al. [21], de esta forma las características de representación de los datos difusos son explícitamente definidos y este tipo de información puede ser fácilmente accedida a través de interfaces genéricas Web para consultar bases de datos.

En un trabajo reciente Ma et al. [88] se centran en técnicas de consulta de descubrimiento de conocimiento utilizando Lógicas de Descripción y ontologías difusas, mejorando la expresividad mediante la introducción de técnicas de consulta conjuntivas. Además, se incluyen consultas sobre ontologías difusas y algunas extensiones flexibles para el lenguaje de consulta SPARQL, estándar de la W3C.

Otros autores se han centrado en el modelamiento de datos XML difusos, su diseño conceptual, y su almacenamiento en bases de datos, utilizando la teoría de distribución de posibilidades. En [86] los autores desarrollaron un modelo de datos UML difuso para diseñar conceptualmente el modelo XML difuso; también investigaron las conversiones formales del modelo UML difuso al modelo XML difuso y la traducción formal del modelo XML difuso a bases de datos relacionales difusas.

El XML se ha convertido en un estándar de hecho para lograr la interoperabilidad de datos entre aplicaciones y servicios Web, es por ello que los SGBD de última generación y los últimos estándares de SQL consideran ya las formas de comunicarse con aplicaciones que usen este modelo, teniendo incorporadas capacidades de exportar o importar datos en formato XML, por lo que, a nuestro juicio, no se considera necesario modelos difusos en XML.

Vamos ahora a enfocarnos en la utilización del Modelo Objeto-Relacional ya que consideramos que éste ha sido una evolución del modelo Relacional que ha tenido más éxito que los Modelo Orientados a Objeto puros. La adopción de este modelo por varios gestores de bases de datos de reconocido prestigio es lo que da origen a los SGBDOR, los cuales estudiaremos con más detalle a continuación así como las implementaciones del modelo difuso que han tenido éxito utilizando este tipo de gestores.

Un SGBDOR es un Gestor de Bases de Datos que acepta en una capa de abstracción externa la utilización del paradigma orientado a objeto, mientras que en su capa interna gestiona los datos a través de tablas mediante el conocido y eficiente modelo relacional. Como ya se mencionó este modelo permite aprovechar las ventajas de los predecesores creando una nueva área de investigación en bases de datos, que ha sido muy poco estudiada para sistemas que gestionen datos imperfectos utilizando bases de datos difusas.

Cuevas [41] plantea el desarrollo de un sistema de bases de datos difusas orientadas a objetos y su implementación en un gestor distribuido como software libre con características objeto-relacional (PostgreSQL). El modelo difuso propuesto es una extensión fundamentada en los aportes de Marín [90] y Vila et al. [155]. Desarrolló el modelo teórico de Bases de Datos Difusas orientadas a objetos incorporando las extensiones pertinentes; diseñó su aplicación utilizando el SGBDOR PostgreSQL y finalmente implementó una interfaz de usuario para el acceso amigable a las extensiones difusas incorporadas denominada pg4DB y mostrada en Cuevas et al. [42]. Además presentó la personalización de operadores de comparación difusos por parte del usuario con el concepto de perspectivas de comparación para objetos difusos y dominios conjuntivos.

Esta investigación utiliza el paradigma objeto relacional de bases de datos y representa un antecedente directo en cuanto al modelo difuso que implementa.

Se presenta una forma de flexibilización en la utilización de operadores de comparación difusos con el concepto propuesto de perspectivas de comparación para objetos difusos y dominios conjuntivos. Lo más importante es que proporciona un adecuado fundamento teórico para el modelo a utilizar en la presente propuesta, pero que tiene limitaciones por su implementación en PostgreSQL el cual sólo implementa ciertas capacidades del paradigma objeto relacional.

Cuevas et al. [42] proponen pg4DB como un sistema que permite la gestión de objetos difusos en un SGBDOR; su interfaz permite a los usuarios crear objetos imprecisos y operar con ellos utilizando sentencias estándares del SQL objeto-relacional. Aunque pg4DB se construye sobre PostgreSQL, los fundamentos teóricos pueden ser usados para extender otros sistemas objeto-relacionales de manera similar. Esta interfaz permite dar amigabilidad al usuario para la gestión del modelo de bases de datos difusa propuesto por los autores, utilizando PostgreSQL.

Este artículo también es de sumo interés porque representa otro antecedente directo de la presente investigación, debido al diseño del modelo difuso empleado, las perspectivas de comparación para objetos difusos y dominios conjuntivos, cómo se hace la implementación en un SGBDOR, así como la interfaz realizada para proporcionar facilidades en la gestión de datos difusos al usuario. La ventaja de la propuesta del presente autor está en recurrir a un gestor como el Oracle que tiene mayores ventajas reconocidas en la utilización del modelo objeto relacional, además de proponer el diseño de un modelo difuso de semánticas adaptables basadas en el contexto para definir las preferencias del usuario.

La tesis doctoral de Barranco [12] también se centra en establecer las bases para permitir el manejo de información imperfecta en sistemas gestores de bases de datos objeto-relacional utilizando los mecanismos de la lógica difusa (FORDBMS). Propone un modelo de bases de datos que formaliza las características del FORDBMS, además herramientas como el álgebra para el acceso y manipulación de la información imperfecta. Luego de un estudio de las implementaciones precedentes efectuó una implementación utilizando el SGBDOR Oracle.

Medina et al. [100] exponen los cambios de las estructuras y métodos en el Sistema Gestor de Bases de Datos Objeto-Relacional Difuso (FORDBMS) Oracle para proporcionar al diseñador los mecanismos necesarios para establecer el comportamiento deseado de operadores de comparación difusos, de acuerdo a las semánticas de la aplicación del usuario. El FORDBMS apoya los cambios a través de la implementación de un catálogo que almacena parámetros que definen el comportamiento de los comparadores, además se define e implementan los

tipos, métodos y operadores que proporcionan la funcionalidad. Como ejemplo, los autores antes mencionados presentaron una representación flexible de la estructura de la columna vertebral de seres humanos con escoliosis [69].

Estas investigaciones constituyen un aporte significativo para el presente estudio debido a que se pone de manifiesto la importancia de la flexibilización para el usuario, en este caso permitiendo operaciones de comparación de acuerdo a la semántica de la aplicación; además de la utilización de un gestor de Bases de datos Objeto Relacional para la implementación de la bases de datos difusa. De nuevo uno de los principales inconvenientes es la complejidad de su implementación además de estar muy ligado a lo que el diseñador del sistema pueda establecer.

La importancia para la actual investigación de estos últimos trabajos es la utilización de Oracle como gestor de bases de datos objeto relacional para la implementación de un modelo de bases de datos difusas y, de acuerdo a nuestro conocimiento, representa uno de los trabajos que utilizan este paradigma de bases de datos, de allí la importancia de su estudio para esta investigación. Sin embargo, nuestra propuesta se enfoca en aprovechar los aportes de la Interacción Persona-Ordenador en el problema del contexto y adaptarlo a un ambiente de representación de datos imperfectos utilizando un gestor de bases de datos objeto relacional difuso, significando un paso adicional hacia una interacción con el usuario más realista y amigable

Debido a que es un antecedente directo y de primordial interés en nuestra investigación en la próxima sección se estudian con más detalle los aportes de Marín [90], Cuevas [41], Cuevas et al. [42] y Vila et al. [155]

3.1.5. El modelo de datos de Objetos Difusos de referencia

A continuación se presenta un resumen de las características principales del modelo de datos de objeto difusos propuesto por Cuevas [41] en su disertación doctoral basado en los aportes de Marín [90] y Vila et al. [155], de interés primordial en esta investigación porque es en el que se inspira el modelo orientado a objeto difusos diseñado en la propuesta para gestionar datos imperfectos.

De acuerdo a lo ya estudiado, en una bases de datos difusa se puede encontrar vaguedad en varios niveles, a saber: valores de los atributos, entidades o clases, las interrelaciones entre clases (incluyendo herencia) y definiciones de tipos flexibles. El modelo de Cuevas [41] se centra en el nivel de valores de los atributos y definición de tipos flexibles, ya que los considera los más importantes

desde un punto de vista práctico. Nuestro trabajo se centra en el primer punto; por este motivo, a continuación, se describe con ejemplos.

A nivel de atributos se permite representar y manipular información imperfecta. El modelo incorpora el uso de dominios difusos con diferente naturaleza que pueden ser usados para representar los atributos de una clase, incorporando el uso de diferentes operadores difusos para cada tipo de dominio. Se puede trabajar con objetos descritos por atributos difusos, así como objetos complejos, definiendo la igualdad entre ellos (en base a criterios distintos), además de operadores para compararlos. Se hace un aporte importante en la flexibilización y personalización de estos operadores a través de las perspectivas de comparación que, si bien no está orientado a la sensibilidad al contexto, puede entenderse como un pequeño paso en esa dirección.

La vaguedad inherente al valor de un atributo está dada por la naturaleza del dominio del atributo, por lo que el modelo de datos que se utilice debe permitir trabajar con diferentes interpretaciones que puedan tener los dominios. Los valores imperfectos en el modelo pueden estar representados por una etiqueta lingüística que dispondrá de diferentes interpretaciones dependiendo de la semántica del dominio donde se define, por un valor perteneciente al universo sobre el que se define el dominio o por un subconjunto difuso.

Partiendo de la cardinalidad se pueden distinguir dos tipos de dominios difusos [41], a saber:

- *Atómicos*: Son aquellos en los cuales el atributo puede tomar un solo valor, donde se pueden tener dominios *sin* o *con* representación semántica asociada. En este último caso (de acuerdo al referencial sobre el que se define) es posible tener dos tipos: referencial finito o referencial continuo (se definen funciones de pertenencia).
- *Conjuntivos*: Permiten que el atributo tome como valor un subconjunto difuso de valores.

Un resumen de los tipos de dominios difusos, dependiendo del valor que puede tomar un atributo presenta en la figura 7 [41].

A partir de este momento se van a presentar ejemplos de dominios difusos definidos sobre un referencial de valores, de acuerdo al esquema propuesto anteriormente.

a) Dominios atómicos con Representación Semántica asociada

En estos dominios la interpretación del valor es monovaluada, pudiendo ser una etiqueta lingüística tomada de un conjunto representadas por distribuciones de posibilidad (referencial continuo) o un valor perteneciente a un

dominio básico (referencial finito); nunca podrá ser una combinación de estos valores. A continuación un ejemplo de cada uno de ellos tomados de Cuevas [41].

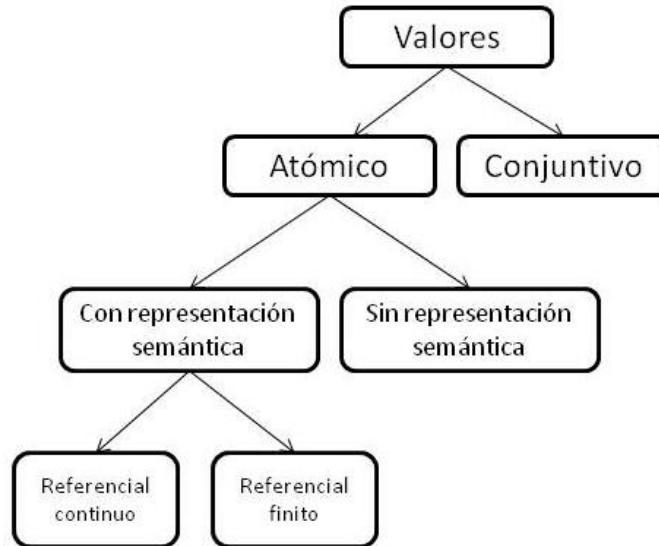


Figura 7. Dominios difusos para valores de atributos (Cuevas [41])

Ejemplo 6. *Dominios atómicos con representación semántica asociada definidos sobre un referencial finito.*

La dureza de un mineral se mide por medio de la escala de dureza de Mohs, la cual tiene un rango entre 1 y 10, donde los números bajos se usan para minerales blandos y los altos para minerales duros, logrando clasificarlos por sus características físicas, en vez de por su composición química. Por ejemplo cuando se dice “el rubí es una piedra muy dura”, se está en presencia de una expresión con términos imprecisos. El dominio básico de la dureza es el conjunto finito $B = \{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10\}$ y se pueden tener el conjunto de valores imprecisos {"Muy duro", "duro", "medio", "blando", "muy blando"}, con las siguientes distribuciones de posibilidad:

$$\text{Muy duro} = \{0.3/7, 0.5/7.5, 0.7/8, 0.8/8.5, 1.0/9, 1.0/9.5, 1.0/10\}$$

$$\text{Duro} = \{0.3/6, 0.6/6.5, 1.0/7, 1.0/7.5, 1.0/8, 1.0/8.5, 0.7/9, 0.5/9.5, 0.2/10\}$$

$$\text{Medio} = \{0.2/2.5, 0.4/3, 0.6/3.5, 0.8/4, 1.0/4.5, 1.0/5, 1.0/5.5, 1.0/6, 1.0/6.5, 0.8/7, 0.6/7.5, 0.4/8, 0.2/8.5\}$$

$$\text{Blando} = \{0.4/1.5, 0.8/2, 1.0/2.5, 1.0/3, 1.0/3.5, 1.0/4, 0.7/4.5, 0.4/5\}$$

$$\text{Muy Blando} = \{1.0/1, 1.0/1.5, 1.0/2, 0.8/2.5, 0.6/3, 0.5/3.5, 0.3/4\}$$

En el momento de determinar el grado de semejanza entre dos valores cualesquiera del dominio, se utiliza la ecuación 3.1 [90].

$$\mu_S(x, y) = \begin{cases} 1 & (x = y) \wedge (x, y \in B) \\ 0 & (x \neq y) \wedge (x, y \in B) \\ \mu_l(z) & ((x = l \in L) \wedge (y = z \in B)) \vee \\ & ((y = l \in L) \wedge (x = z \in B)) \\ \max_{z \in B} \otimes (\mu_x(z), \mu_y(z)) & \text{otro caso} \end{cases} \quad (3.1)$$

donde B es el dominio básico y L el conjunto de etiquetas, l representa una etiqueta lingüística; el dominio real está definido por la unión: $D = B \cup L$; además $x, y, z \in D$. Esta relación de semejanza da el valor del operador “posiblemente igual difuso” (FEQ) definida sobre un dominio con estas características.

Ejemplo 7. *Dominios atómicos con representación semántica asociada definidos sobre un referencial continuo.*

En el caso de la estatura de una persona (en centímetros) el dominio básico es el conjunto de número enteros, donde se pueden definir etiquetas lingüísticas que permiten expresar la imprecisión. En este caso las etiquetas estarán expresadas por medio de funciones de pertenencia trapezoidales, tal como: Bajo = {0, 0, 150, 160}, Promedio = {150, 160, 170, 180} y Alto = {170, 180, 300, 300}. Las funciones de pertenencia se muestran en la figura 8.

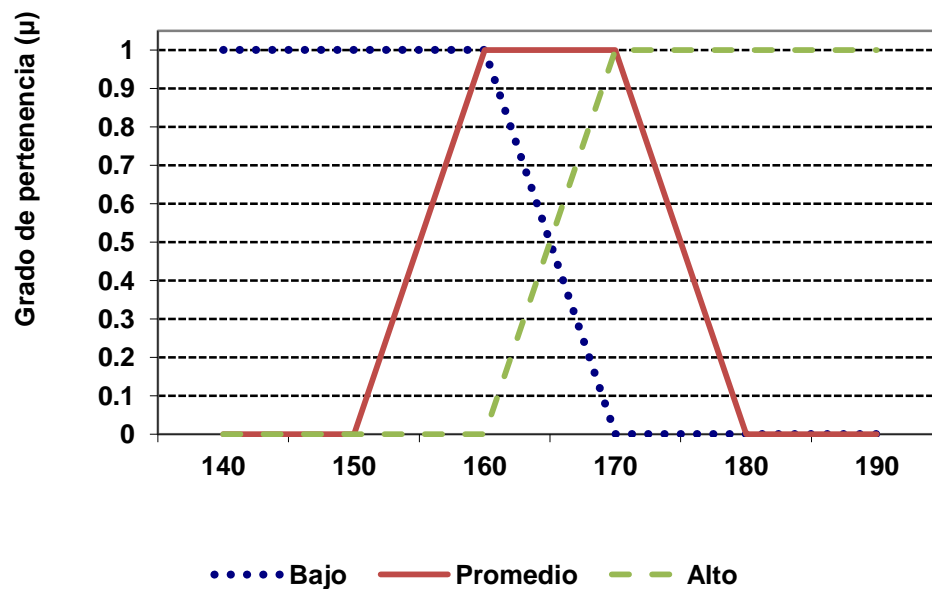


Figura 8. Etiquetas lingüísticas para el atributo estatura

Ejemplo 8. *Dominios atómicos con representación semántica asociada definidos sobre un referencial continuo definido por funciones.*

Esta forma de representación puede ser utilizada cuando no sea adecuado una definición extensiva de la distribución de posibilidad de las etiquetas lingüísticas, por lo que se hace a través de funciones. Se puede utilizar cualquier tipo de función, pero debido a su sencillez de manipulación desde el punto de vista computacional, se estilan lineales (rectas) que representan funciones de pertenencia (FP) trapezoidales (o triangulares) simétricas o no, lo cual es aceptado como suficiente en la comunidad de investigadores de bases de datos difusas, tal como lo reporta Cuevas [41].

La FP es descrita por cuatro parámetros, donde el primer valor indica el inicio del soporte (ω), el segundo valor el inicio del núcleo ($Y1$), el tercer valor el fin del núcleo ($Y2$) y el cuarto valor el fin del soporte (δ). Así el valor del grado de pertenencia para una etiqueta θ en un valor x del dominio subyacente, se calcula según la ecuación 3.2.

$$\mu_{\theta}(x) = \begin{cases} 0 & \text{para } x \leq \omega, \\ \frac{x - \omega}{Y1 - \omega} & \text{para } \omega < x < Y1 \\ 1 & \text{para } Y1 \leq x \leq Y2 \\ \frac{\delta - x}{\delta - Y2} & \text{para } Y2 < x < \delta \\ 0 & \text{para } x \geq \delta \end{cases} \quad (3.2)$$

Una función de pertenencia para la etiqueta “maduro” de la variable lingüística edad se muestra en la figura 9.

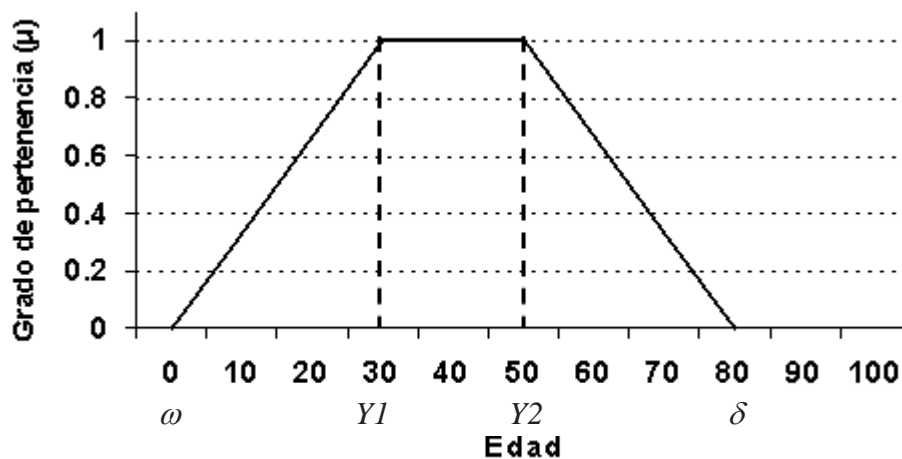


Figura 9. FP de la etiqueta lingüística “maduro”

A continuación se presentan otros tipos de trapezoides (no cerrados) para las etiquetas “viejo” y “joven” de la variable lingüística edad, el grado de pertenencia para una etiqueta θ en un valor x del dominio subyacente, se calcula según las ecuación 3.3 y 3.4, con sus respectivas FP en las figuras 10 y 11.

$$\mu_{\theta}(x) = \begin{cases} 0 & \text{para } x \leq \omega, \\ \frac{x - \omega}{Y - \omega} & \text{para } \omega < x < Y \\ 1 & \text{para } x \geq Y \end{cases} \quad (3.3)$$

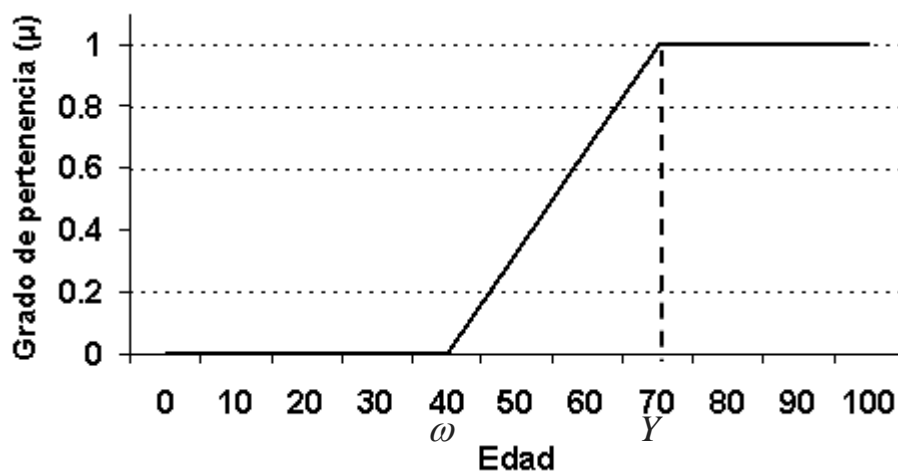


Figura 10. FP de la etiqueta lingüística “viejo”

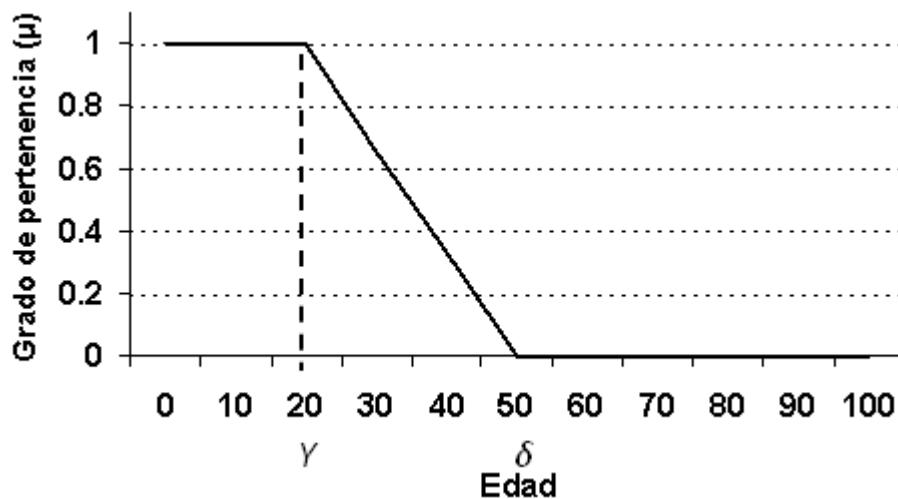


Figura 11. FP de la etiqueta lingüística “joven”

$$\mu_{\theta}(x) = \begin{cases} 1 & \text{para } x \leq Y, \\ \frac{\delta - x}{\delta - Y} & \text{para } Y < x < \delta \\ 0 & \text{para } x \geq \delta \end{cases} \quad (3.4)$$

En el caso de representaciones triangulares simétricas, pueden usarse para representar números difusos “aproximadamente X” o “alrededor de X”, para esta función se usan dos parámetros, el valor Y que es el núcleo, y el valor β que indica el margen de desplazamiento en Y (hacia la izquierda y hacia la derecha) para formar el soporte. Si no es simétrica, se necesitan tres parámetros inicio del soporte (ω), el núcleo (Y), y el fin del soporte (δ).

A continuación se presenta la ecuación 3.5 que representa a esta última y su respectiva FP en la figura 12.

$$\mu_{\theta}(x) = \begin{cases} 0 & \text{para } x \leq \omega, \\ \frac{x - \omega}{Y - \omega} & \text{para } \omega < x < Y \\ 1 & \text{para } x = Y \\ \frac{\delta - x}{\delta - Y} & \text{para } Y < x < \delta \\ 0 & \text{para } x \geq \delta \end{cases} \quad (3.5)$$

La semejanza entre dos valores del dominio, debe ser determinado por expresiones apropiadas. En el caso de comparar un valor x del dominio básico con una de las etiquetas $l_i \in L$, una de las formas es recurrir a la correspondiente ecuación de la distribución de posibilidad, ver ecuación 3.6.

$$\mu_S(x, l_i) = \mu_{l_i}(x) \quad (3.6)$$

Si se desea determinar la semejanza entre dos etiquetas lingüísticas l_1 y l_2 , tal que $l_1, l_2 \in L$ se utilizan también las distribuciones de posibilidad de las etiquetas mediante la siguiente ecuación 3.7.

$$\mu_S(l_1, l_2) = \sup \min(\mu_{l_1}(x), \mu_{l_2}(x)) \quad (3.7)$$

Por medio de estas ecuaciones se puede dar soporte al operador de “posiblemente igual difuso” (FEQ) para este tipo de dominio. Debido a que existe una relación de orden también es posible definir otros operadores de posibilidad difusos tales como: mayor difuso (FGT), mayor o igual difuso ($FGEQ$), menor difuso (FLT), menor o igual difuso ($FLEQ$), mucho mayor difuso (MGT) o mucho menor difuso (MLT) que pueden ser revisados en Cuevas [41].

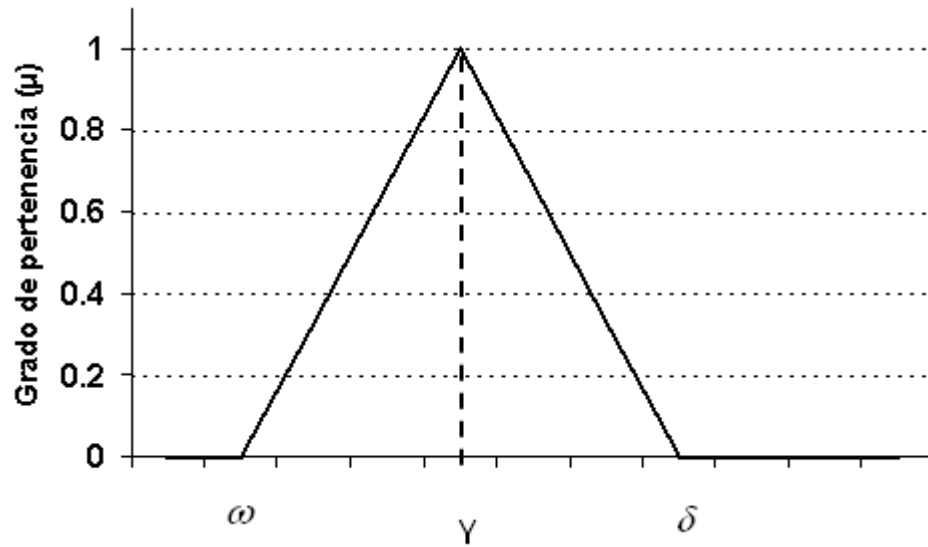


Figura 12. FP triangular para la función “alrededor de Y ”

b) Dominios atómicos sin Representación Semántica asociada

Para este tipo de datos se define un dominio básico B formado por un conjunto de etiquetas lingüísticas. Dado que no existen distribuciones de posibilidad asociadas a las etiquetas, se utiliza una relación de semejanza sobre B que permitirá realizar comparaciones entre dos valores del dominio. Para calcular la semejanza entre dos etiquetas sólo es necesario observarla en una tabla definida entre ellos; así se podrá dar soporte al operador FEQ para determinar la semejanza entre dos valores.

Ejemplo 9. *Calidad de un producto.*

Para este caso se definen etiquetas lingüísticas tales como: *bueno*, *media* y *baja*; que no tienen asociado distribuciones de posibilidad alguna, por lo que se debe definir una relación de semejanza como la mostrada en la tabla 1.

Tabla 1. Relación de semejanza del atributo *calidad*

	<i>Alta</i>	<i>Media</i>	<i>Baja</i>
<i>Alta</i>	1	0.6	0.1
<i>Media</i>		1	0.4
<i>Baja</i>			1

En caso de aumentar el dominio con nuevas etiquetas lingüísticas, el nuevo valor debe estar representado por un subconjunto difuso de las etiquetas previamente definidas para el dominio, a partir de allí se modifica la relación de semejanza mediante la ecuación 3.8.

$$\forall x \in D, \mu_S(l, x) = \max_{y \in D} (\mu_S(y, x) \otimes \mu_L(y)) \quad (3.8)$$

donde D es el conjunto de etiquetas,
 S la relación de semejanza previamente definida,
 L un subconjunto difuso sobre D que define la nueva etiqueta.

c) Dominios conjuntivos sobre valores

En este caso el valor del atributo no es una distribución de posibilidad, sino que son formados por una combinación de todos los posibles valores del referencial.

Partiendo de la existencia de un dominio básico B , los valores que puede tomar un atributo definido sobre este dominio será un subconjunto A con una función de pertenencia representada por la ecuación 3.9.

$$\forall A \text{ subconjunto difuso de } B \mid \text{Soporte}(A) \text{ es finito, } A \subseteq \mathcal{P}(B \times [0,1]) \quad (3.9)$$

Por lo que cualquier valor para este dominio será un subconjunto difuso que tendrá una función de pertenencia de soporte finito representada por un conjunto de pares de valores, en los que el primer valor pertenece al dominio básico B y el segundo un número real en el intervalo $[0,1]$.

Si se tiene D como el conjunto de todos los posibles valores de un atributo $D = x_1, x_2, \dots, x_n$, el valor del atributo att_1 definido sobre un dominio conjuntivo con referencial de valores se representa de la siguiente forma de acuerdo a la ecuación 3.10 [41].

$$att_1 = \mu att_1(x_1) / x_1 + \mu att_1(x_2) / x_2 + \dots + \mu att_1(x_n) / x_n \quad (3.10)$$

donde el operador $+$ es interpretado como un *and* lógico.

Para determinar la igualdad entre dos valores de este dominio se debe recurrir al cálculo del parecido entre dos conjuntos difusos, lo cual suele hacerse en términos de inclusión doble, mostrados en la ecuación 3.11.

$$A = B \text{ si y sólo si } (A \subseteq B) \wedge (B \subseteq A) \quad (3.11)$$

Para el cálculo de la inclusión se tomó la propuesta de Marín [90] mostrada en la ecuación 3.12.

$$N(A|B) = \min\{I(\mu_A(x), \mu_B(x))\}, \text{ donde } x \in U \quad (3.12)$$

Finalmente para el cálculo de semejanza entre los dos conjuntos deben ser combinados los dos grados de inclusión, como se presenta en la ecuación 3.13.

$$\mu_S(A, B) = OC(N(B|A), N(A|B)) \quad (3.13)$$

donde OC es un operador de combinación, entre los cuales puede escogerse: una *t-norma*, *t-conorma* o el promedio para combinar los grados de inclusión; la primera es la más restrictiva.

Ejemplo 10. *Idiomas de un investigador.*

Un investigador puede dominar varios idiomas y cada uno con un determinado nivel que varía de idioma en idioma. En este caso el dominio básico es el conjunto de idiomas $\{\text{inglés}, \text{francés}, \text{ruso}, \text{italiano}, \text{español}\}$; un investigador1 domina los idiomas descritos por el subconjunto difuso $\{\text{Español}/1.0 + \text{Inglés}/0.6 + \text{Francés}/0.4\}$ mientras que el investigador2 domina los idiomas $\{\text{Español}/1.0 + \text{Inglés}/0.7 + \text{Francés}/0.3\}$. Si se desea saber si *hablan los mismos idiomas*, se tiene que hacer de la siguiente forma:

$$\mu_s(\text{inv1.idioma}, \text{inv2.idioma}) = \otimes(\min(1, 1, 0.75), \min(1, 0.85, 1)) = 0.75$$

Para los cálculos se utilizó como *t-norma* el mínimo, *t-conorma* el máximo y la implicación de Gödel los cuales son los más ampliamente usados de acuerdo a lo planteado por Cuevas [41].

Es de hacer notar que para la presente investigación, este marco teórico del modelo de datos difusos con la operación de comparación *FEQ* entre los diferentes dominios para valores de atributos difusos es suficiente. Sin embargo en Cuevas [41]; Marín et al. [91] y Marín [90] se puede hacer una revisión más profunda para el caso de la representación de objetos difusos (incluyendo los complejos), formas de comparación entre objetos con atributos difusos, perspectivas de comparación (objetos difusos y dominios conjuntivos) y el tratamiento de recursividad en objetos complejos; así como un sistema de reglas difusas para inferir valor de atributos.

3.4. Discusión

Una de las características del pensamiento humano es la habilidad de resumir información en etiquetas, lo cual proporciona información imperfecta que no es reflejada en sistemas de información tradicionales. La utilización de conjuntos difusos puede ayudar a precisar este lenguaje y representar este tipo de información imperfecta, específicamente en sistemas gestores de bases de datos. La importancia de modelar este tipo de información es porque representa cuantiosa información sobre la cual se basan las decisiones del ser humano.

Además, como hemos visto, el significado dado a estas etiquetas es subjetivo, de allí la conveniencia de utilizar teorías postuladas por Zadeh (posibilidad, computación con palabras y computacional de percepciones). Por otro lado el significado de los términos descriptivos depende en gran medida del contexto en que son utilizados, cuestión que no puede ser reflejada en sistemas tradicionales y a la que se le ha prestado una atención limitada en la literatura

Por estas razones es de vital importancia el estudio elaborado de los modelos de datos difusos y sus diversas implementaciones utilizando los

paradigmas Relacional, Orientado a Objetos y Objeto-Relacional. Nuestra propuesta aborda la problemática de la representación de información imperfecta relacionándola con el contexto del usuario, para producir sistemas personalizados

Decidimos utilizar un modelo objeto-relacional, debido al éxito que han tenido los antecedentes utilizando dicho modelo, representando en forma más efectiva datos proporcionados por el mundo real, personalizando el resultado de acuerdo al contexto basado en preferencias del usuario.

Mediante la utilización del paradigma orientado a objetos presentaremos un modelo de datos para representar dominios difusos que sirva de base para su implementación de una manera transparente en SGBDOR, además la definición del modelo de contexto a utilizar en un esquema de interacción de un usuario con bases de datos nos permitirá crear las bases para el desarrollo de aplicaciones de bases de datos difusas sensibles al contexto.

Es por ello que en el próximo capítulo presentamos los fundamentos de nuestra propuesta con un Modelo para Sistemas de Información Sensibles al Contexto presentando: un esquema de interacción de un usuario con bases de datos, la definición del modelo de contexto en dicho ambiente y una arquitectura para el desarrollo de sistemas como el propuesto.

Capítulo 4

Sistema de BD Sensible al Contexto

Como hemos visto en capítulos previos de esta memoria el contexto tiene un impacto significativo en la manera que los seres humanos actúan y en cómo interpretan y perciben las cosas del mundo real. En este sentido a través de la presente investigación, se ha establecido la necesidad de realizar aplicaciones que sean más cercanas a la perspectiva humana haciéndolas sensibles al contexto.

Frente al paradigma tradicional de operación de un Sistema de Bases de Datos donde el usuario proporciona una entrada (ingreso de datos o consulta), se procesa y se obtiene la salida, con la certeza de que siempre que la misma entrada tendrá siempre iguales resultados como salida (paradigma de coincidencia exacta); se propone un enfoque flexible de sensibilización al contexto, donde dada la entrada y el contexto del usuario que proporciona tal entrada, se efectúe el proceso y se puedan obtener resultados diferentes (personalizados) que van a depender no sólo de los datos ingresados sino también del contexto del usuario que interacciona con las bases de datos difusa.

Corresponde en este capítulo mostrar el Modelo de Bases de Datos Difusas Sensibles al Contexto para el desarrollo de sistemas de bases de datos inteligentes basándonos en el modelado orientado a objetos de datos contextuales. Este modelo nos permitirá realizar consultas sensibles al contexto que puedan afectar tanto los resultados como la presentación de los mismos.

Comenzamos con definir el esquema de interacción de un usuario con un Sistema de Bases de Datos Sensible al Contexto (SBDSC), luego definimos el modelo de Contexto en nuestra propuesta, estudiamos la sensibilidad al contexto a la salida del sistema, para luego presentar nuestra propuesta de una Arquitectura del SBDSC.

4.1. Esquema de Interacción

Vamos a empezar nuestra propuesta analizando el esquema general de interacción de un usuario con el Sistema de Bases de Datos Sensible al Contexto (SBDSC). Si nos centramos en la interacción del usuario con el sistema, desde un punto de vista muy general, podemos considerar que el sistema debe responder a una instrucción indicada por el usuario, ofreciendo como resultado una respuesta satisfactoria. Si el sistema es *sensible al contexto*, la entrada que se debe considerar trasciende a la propia instrucción, a la que debe añadirse información relativa al contexto en el que se ha planteado.

En la figura 13 se presenta el esquema general de interacción de un usuario con un SBDSC. Como puede observarse, el sistema debe tener en cuenta además de la entrada tradicional del sistema (la instrucción que indica el usuario), información relativa al contexto en el que se plantea dicha entrada incluyendo tanto información adicional aportada por el propio usuario como información tomada del ambiente. El procesamiento que realiza el sistema y la salida que se ofrece como resultado de dicho procesamiento deben adecuarse para responder a la cuestión planteada teniendo en cuenta el contexto en el que se plantea. La semántica de la salida, el conjunto de datos que la conforman y su presentación deben adaptarse de forma apropiada.

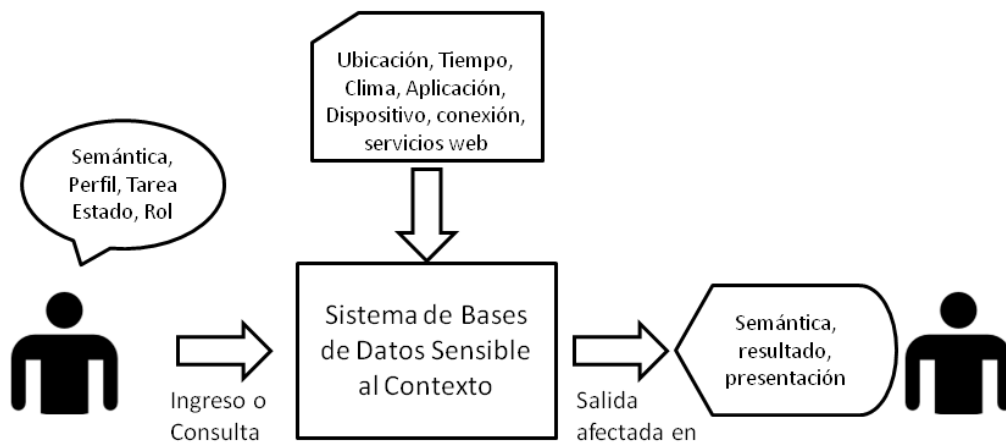


Figura 13. Esquema de interacción de un usuario con un Sistema de Bases de Datos Sensible al Contexto

Veamos con detalle dos de los elementos principales de dicho esquema de interacción: el contexto y la salida adaptada al contexto.

4.2. El Contexto

El contexto se ha utilizado ampliamente como un concepto para dar significado a las palabras en la lingüística y para la comprensión de las circunstancias que rodean un evento particular [6]. En los capítulos anteriores se ha hecho un amplio repaso en relación al concepto de contexto y su aplicación en el mundo de la Computación.

Nuestra idea no es desarrollar un modelo de contexto universal que recoja todas las posibles definiciones de contexto en el ámbito de los Sistemas de Información, sino proponer un modelo flexible que pueda ser adaptado de acuerdo al interés particular de cada dominio de aplicación, lo cual puede ser proporcionado mediante la utilización del modelado de datos orientado a objetos. Formulamos pues, un modelo conceptual representado bajo este paradigma.

De acuerdo a lo revisado anteriormente en las definiciones de contexto, dado que estamos en un ámbito donde se puede obtener información de contexto que sea explícitamente indicada por el usuario, inferida por el sistema u obtenida por sensores de ambiente, en la figura 14 se presenta un diagrama UML del modelo del contexto que proponemos en un esquema de interacción con sistemas de bases de datos.

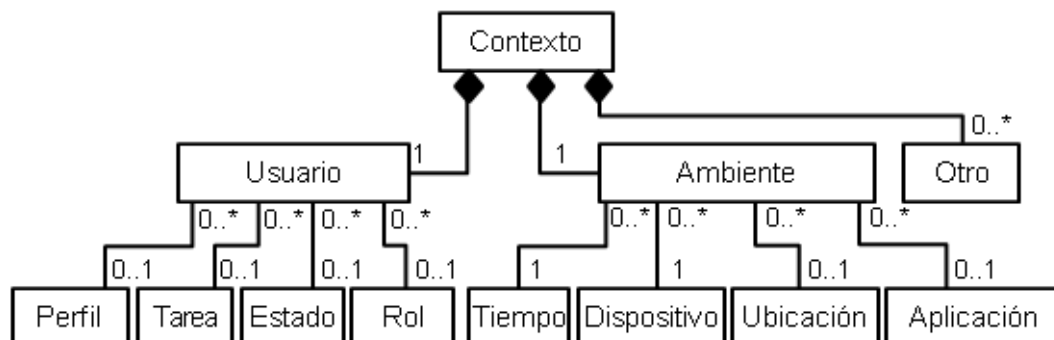


Figura 14. Modelo de contexto en un esquema de interacción Usuario con SBDSC

Como se puede observar, el modelo de contexto propuesto está compuesto por diversos elementos entre los que están:

- Usuario: con información propia e información relativa a su perfil, la tarea que está realizando, su estado o el rol que desempeña.
- Ambiente: datos ambientales del momento en que se hace la interacción tales como tiempo, dispositivo, ubicación y aplicación.
- Otros: cualquier otra información contextual que el administrador del sistema requiera definir dependiendo del dominio de la aplicación.

De una manera más formal, podemos establecer la siguiente definición.

Definición 3. *Contexto de interacción entre un usuario y un SBDSC.*

El Contexto de interacción entre un usuario y un SBDSC es una 3-tupla:

$$\text{Contexto} = \langle u, a, o \rangle$$

donde:

- *u es una 4-tupla Usuario = $\langle p, t, e, r \rangle \in P \times T \times E \times R$ y*
 - *P es el conjunto de Perfiles*
 - *T es el conjunto de Tareas*
 - *E es el conjunto de Estados*
 - *R es el conjunto de Roles*

- *a es una 4-tupla Ambiente = $\langle tp, d, u, a \rangle \in TP \times D \times U \times A$ y*
 - *TP es el conjunto de instantes de tiempo*
 - *D es el conjunto de dispositivos de conexión*
 - *U es el conjunto de ubicaciones*
 - *A es el conjunto de programas de aplicación desde los que se puede hacer la interacción.*

- *o es una n-tupla que recoge tantas componentes como aspectos contextuales adicionales distintos a las anteriormente definidas se quieran considerar.*

De la anterior definición, se entiende que un *contexto* es una instancia de un *modelo de contexto* basado en las componentes de Usuario, Ambiente y Otros.

En nuestro análisis del contexto vamos a considerar dos grandes grupos de aspectos: aquellos que *girán* alrededor del usuario y aquellos relativos al ambiente, entre otros adicionales. Veámoslos con detalle.

4.2.1. Contexto relativo al usuario

El usuario de un sistema de bases de datos debe estar siempre identificado debido a los niveles de seguridad que garantizan que los datos puedan ser accedidos sólo por aquellos que tengan privilegio para hacerlo. Por este motivo, un sistema de bases de datos siempre tiene una primera información del contexto que es la identificación del usuario que interacciona con el mismo. Al propio usuario le acompañan otros muchos aspectos que deben ser considerados a la hora de adaptar la respuesta del sistema:

- Perfil

Cuando el usuario se conecta al sistema puede tener asociado un perfil en el que se aporten datos relativos al usuario y sus preferencias. Este perfil de usuario permitirá dar relevancia, filtrar y personalizar las respuestas del sistema.

- Tarea

Con este aspecto se quiere identificar qué tipo de actividad está haciendo el usuario cuando efectúa la consulta en el marco del dominio de aplicación en el que está implantado el sistema (por ejemplo caminando, haciendo ejercicios, en bicicleta, manejando, efectuando una operación quirúrgica, en cama, entre otros).

Aunque hemos considerado este aspecto como parte del contexto aportada por el usuario, algunos dispositivos inteligentes modernos están equipados con una amplia variedad de sensores que se pueden utilizar para determinar la tarea en algunos dominios de aplicación. Por ejemplo, mediante sistemas de posicionamiento global (GPS) y acelerómetros, se puede determinar si un usuario está quieto, caminando, corriendo o en un vehículo, en el momento de la interacción [133].

Otro ejemplo, en el ámbito de salud, es identificar la tarea que está realizando un médico especialista cuando realiza la interacción con el sistema: si está realizando una operación quirúrgica, está en su despacho haciendo consultas a pacientes o está de descanso. Esto puede ayudar a determinar la relevancia de posibles respuestas, su precisión y su forma de presentación, por ejemplo.

- Estado (físico, emocional, social)

El estado actual del usuario ya sea en lo físico, emocional o social, también va a ser relevante para permitirnos ajustar los resultados de las consultas y su forma de presentación, así como la semántica de los conceptos utilizados.

De nuevo, aunque lo hemos colocado como explícito, en determinados dominios, también se pueden utilizar sensores que determinen el estado de forma dinámica. Por ejemplo, hoy en día, los teléfonos inteligentes permiten monitorizar la frecuencia cardíaca, ofreciendo información relacionada con el estado emocional y la salud de la persona.

En cuanto al estado emocional o de ánimo, la primera aproximación en nuestro modelo es que el usuario (paciente en el caso del hospital) diga explícitamente su estado de ánimo (ira, alegría, tristeza, miedo, disgusto y sorpresa, entre otros). En áreas como la de Interacción Persona-Ordenador se han utilizado diferentes técnicas para reconocer las emociones humanas, donde algunas propuestas se han concentrado en las expresiones faciales y el

lenguaje, otras en parámetros fisiológicos como la frecuencia cardíaca y la actividad electro-dérmica [128]; también se ha tratado de prever el comportamiento de una persona utilizando vídeos y sensores, mediante técnicas de percepción inteligente [108]. Además, dentro de Inteligencia Ambiental, la computación afectiva estudia como habilitar a las computadoras para reconocer las emociones humanas y para actuar en consecuencia [128].

En cuanto al estado social, puede ser importante conocer las personas que rodean al usuario (está sola, acompañada con parientes o amigos) o si, por ejemplo, está inmerso en el tráfico de una ciudad. También pueden considerarse cuestiones relativas a su estatus económico.

- Rol

Un mismo usuario puede asumir en la base de datos diferentes roles. En muchos sistemas de bases de datos, el rol establece una forma intermedia de asociar privilegios a usuarios de una forma mediada, asociando los privilegios a los roles, que los usuarios pueden asumir. Es decir, mediante un rol se pueden agrupar muchos usuarios para asociarles un perfil del grupo. El rol, por tanto, puede ser considerado como otro elemento del contexto directamente asociado al usuario.

De nuevo, en el ámbito hospitalario, podemos tener cientos de usuarios, que se pueden agrupar los mismos por médicos referentes, médicos evaluadores, terapeutas físicos (fisioterapeutas), enfermeros, administradores del centro y familiares de pacientes, entre otros.

4.2.2. Contexto relativo al ambiente

El sistema de bases de datos también debe ser consciente de aspectos dinámicos del ambiente que puedan afectar la interacción con el sistema de bases de datos sensible al contexto:

- Tiempo

Ya hemos mencionado que hay casos en que se pueden obtener datos a través de sensores que están asociados al tiempo (fecha y hora) y que pueden condicionar el procesamiento y la salida del sistema.

Por ejemplo, en un hospital es importante este aspecto del contexto para cuando el personal médico haga consultas acerca de los tratamientos que deben ser aplicados o el calendario de intervenciones quirúrgicas.

- Dispositivo

En relación con el dispositivo, se puede identificar si el usuario se conecta desde un dispositivo fijo (computador de un despacho), o móvil (teléfono inteligente, tableta). Este contexto permitirá presentar los resultados en forma diferente (resumidos, a través de texto o multimedia). Aplicable a cualquier tipo de consulta en cualquier dominio de aplicación.

- Ubicación

La ubicación puede describir no sólo el lugar físico donde está la persona (ciudad, país, coordenadas) proporcionada por sistemas tales como GPS; además, puede proporcionar información espacial relacionada tal como la velocidad u orientación, incluyendo información de la zona en que se encuentra el usuario. En general los datos obtenidos por este tipo de contexto pueden ser proporcionados (además de sensores) por otras aplicaciones o servicios web de terceros.

- Aplicación

El usuario puede interactuar directamente con el sistema o hacerlo a través de una aplicación. En este caso es importante conocer el tipo de aplicación que está haciendo la consulta.

4.2.3. Otros

En nuestra propuesta dejamos abierta la posibilidad de definir otros aspectos contextuales, que de acuerdo a nuestro estudio no consideramos primordiales en nuestro esquema de interacción. Por ejemplo, desde el punto de vista del ambiente, se puede considerar el *tipo de conexión* a la red, como un factor que también determina la presentación de resultados en diversos medios. Aplicable de nuevo a cualquier tipo de consulta en cualquier dominio de aplicación. Y como éste, muchas otros que pueden ser de interés en el dominio de aplicación.

4.3. Sensibilidad al contexto de la salida del sistema

El contexto influye en los resultados producidos por una consulta; una misma consulta realizada en diferentes contextos puede dar resultados diferentes aún si los datos contenidos en la base de datos no han cambiado. A continuación, se describen los aspectos de la salida de una consulta que consideramos pueden

ser afectados en el esquema de interacción de un usuario con una base de datos debido a cambios en el contexto.

4.3.1. Semántica

Las instrucciones que aporta el usuario en sistemas convencionales están escritas con lenguajes de consulta con una semántica clara y exenta de ambigüedades; nada más lejos de la forma en que los usuarios nos comunicamos con el lenguaje natural, en el que la semántica de muchos términos y expresiones varía de un usuario a otro, de un contexto a otro. Existen muchas propuestas significativas en la literatura de sistemas que han mejorado sus lenguajes de interacción con el usuario en este sentido; en estos sistemas, la semántica de los términos y expresiones que aparecen en las instrucciones que indica el usuario son un elemento fundamental. Los Sistemas de Bases de Datos Difusos son un ejemplo claro de sistemas que han extendido sus forma de interactuar con el usuario basándose en lenguajes más cercanos al lenguaje natural, en este caso, utilizando conceptos y técnicas de la Teoría de Conjuntos Difusos y la Teoría Computacional de Percepciones de Zadeh. En los sistemas difusos convencionales, la semántica del lenguaje es fija y definida por el diseñador, cuando en la realidad debe ser variable y dependiente del contexto. En el capítulo 5 prestaremos especial atención a este elemento primordial a través de una propuesta de manejo adaptable de semánticas en función del contexto, tanto a la hora de *entender* la entrada, como a la hora de *comunicar* la salida.

4.3.2. Resultado

Uno de los objetivos del sistema de bases de datos sensible al contexto es ajustar o personalizar el resultado de acuerdo al contexto para flexibilizar las respuestas que se obtienen de la consulta, abordando los problemas de rigidez de la consulta que se producen con el paradigma de coincidencia exacta. Esto incluye diversas características del resultado que hemos identificado en nuestro modelo, a saber: tamaño, precisión, orden, clasificación y tipo. A continuación se describe el significado de cada uno de ellos.

- Tamaño

Por tamaño entendemos el número de tuplas o de objetos que componen el resultado. Este número puede variar dependiendo del contexto. En cuanto al número de columnas en nuestra propuesta, en principio, no va a ser afectado porque nos proponemos *utilizar* el lenguaje SQL estándar de un SGBDOR y en la

sentencia SELECT de la consulta ya están identificadas las columnas o los atributos que solicita el usuario como salida. Aunque podrían hacerse las mismas consideraciones y tratarse de una manera similar.

El tamaño del resultado puede estar condicionado principalmente por dos factores: la información relativa al contexto condiciona el procesamiento del sistema y cambia el número de elementos que hay en el resultado; o bien, a la hora de presentar el resultado, este se limita en base al contexto.

Un ejemplo de esto último lo tenemos en relación con la dimensión dispositivo; dependiendo del tipo de dispositivo con que se conecte el usuario (computador de escritorio o dispositivo móvil) se puede presentar un número diferente de tuplas (más o menos respectivamente)

Para el caso de que el contexto influya en el procesamiento y por tanto en la cardinalidad del resultado podemos centrarnos en el perfil. En el perfil se pueden definir diferentes parámetros contextuales (como intereses, hábitos, país donde reside, preferencias, entre otros), que al incorporarlos a la consulta funcionan como restricciones adicionales que van a tener un efecto sobre el número de filas que cumplen la condición.

- Precisión

En los sistemas que lo permiten (por ejemplo los que utilizan conceptos de la Teoría de Conjuntos Difusos), la precisión es un aspecto del resultado que se debe manejar y que está influenciado por el contexto.

En el caso del hospital los resultados de una consulta acerca de un diagnóstico de un paciente deben ser más precisos cuando el paciente está en estado grave (por ejemplo, en terapia intensiva) que cuando son para su seguimiento.

- Orden

El orden en que se presentan los resultados es importante cuando se obtienen muchas respuestas y se quiere priorizar o jerarquizar las ajustadas a las necesidades de los usuarios. Es una forma complementaria al tamaño de la respuesta y está directamente relacionada con el ajuste de cada posible resultado con respecto a la instrucción aportada por el usuario y su contexto determinado.

- Formato

La influencia del contexto en el formato de presentación es evidente. Por ejemplo, el tipo de medio o las preferencias de usuario pueden condicionar la forma de presentar los resultados. No sólo en su cuestión más *estética*, sino también en el propio medio de comunicación. Un usuario con cierta discapacidad visual puede recibir los resultados ampliados o incluso en forma de audio.

4.4. Arquitectura del Sistema de Bases de Datos Sensible al Contexto

Hemos prestado atención hasta ahora en este capítulo a dos de los elementos fundamentales del esquema de interacción: el contexto y la salida sensible al contexto. Ahora vamos a centrarnos en cómo debe construirse un sistema de bases de datos para incorporar sensibilidad al contexto. Con ello, aportamos un marco de trabajo para el desarrollo de aplicaciones sensibles al contexto que, en el próximo capítulo, usaremos para trabajar con información imperfecta y términos del lenguaje natural, con el objetivo de que sean sistemas flexibles y personalizados.

4.4.1. Integración de la sensibilidad al Contexto en Sistemas Informáticos

En una arquitectura de alto nivel de un sistema sensible al contexto, el diseño se realiza de acuerdo al dominio de la aplicación; modelando los elementos que afectan el conocimiento, los servicios y las acciones, que deben estar disponibles al usuario en tiempo de ejecución al activarse un contexto.

Como ya hemos indicado en la sección anterior, la información contextual actúa como fuente de datos para cambios relacionados con la entrada y la salida, lo que representa comportamientos alternos proporcionando diferente información a pesar de que los datos del sistema permanecen sin cambios. En un sistema tradicional los datos contextuales no son tratados como información especial y el sistema incluye implícitamente todos los comportamientos diferentes sin tener presente las múltiples facetas que puede tener el ambiente de una aplicación.

En gestión de la información, los sistemas sensibles al contexto están principalmente dedicados a determinar qué porción de la información completa es relevante con respecto a las condiciones ambientales. La adaptación de datos (*data tailoring*) basada en el contexto propuesto por Bolchini y col [18], es una actividad consistente en definir vistas de datos, basada en:

1. La identificación de diversos contextos que la aplicación del usuario va a experimentar en el escenario previsto.
2. El diseño de un conjunto de vistas del usuario para cada uno de los contextos identificados.

El objetivo es apoyar al diseñador de aplicaciones de gestión de datos, ya sean relacionadas con gran cantidad o con pequeñas cantidades de datos (como

plataformas portátiles o sistemas de gestión de datos livianos), para determinar y crear diversas vistas a ser utilizadas en los diferentes contextos, siguiendo un enfoque sistemático.

En la figura 15 se presenta una arquitectura general de un sistema de información sensible al contexto propuesto en [18]; se puede notar que el contexto actual es utilizado para personalizar tanto los datos como las aplicaciones. Hoy en día, la cantidad disponible de datos y fuentes de los mismos no sólo requiere integrarlos, sino también filtrar o adaptar la porción relevante al usuario a fin de:

- a) Proporcionar un conjunto de datos adaptado apropiadamente.
- b) Coincidir con las restricciones físicas de los dispositivos.
- c) Operar sobre una cantidad de datos manipulable (para mejorar la eficiencia de procesamiento de consultas).
- d) Facilitar al usuario datos relevantes al tiempo y ubicación (aplicaciones móviles).

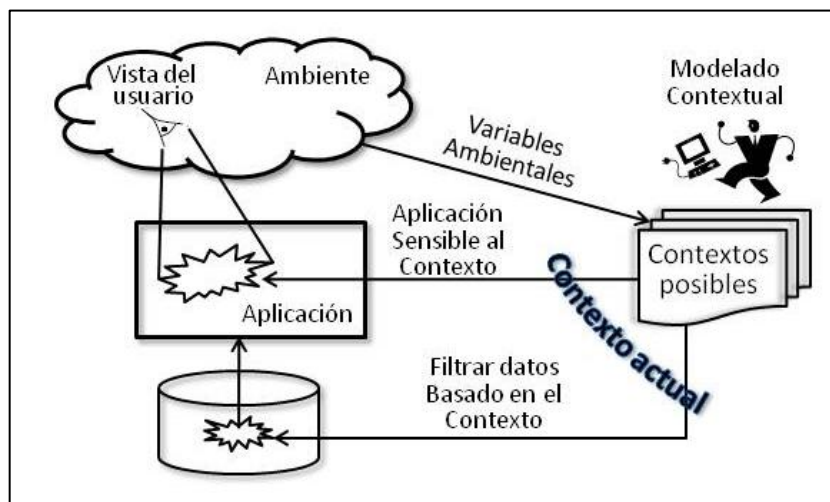


Figura 15. Arquitectura de un Sistema Sensible al Contexto [18]

En un más alto nivel de abstracción que lo proporcionado normalmente en sistemas sensibles al contexto desarrollados en el área de computación ubicua, se necesita gestionar información proporcionada por el ser humano, que como hemos visto utiliza un lenguaje natural que no puede ser reflejado adecuadamente con la utilización de sistemas tradicionales, de allí la necesidad de comenzar por un modelo inspirado en la arquitectura de tres capas estandarizada (ANSI/SPARC) para sistemas de bases de datos.

En la figura 16 se presenta el modelo contextual por capas propuesto por Schmidt [132], el cual describe tres niveles de abstracción en la arquitectura de un sistema de gestión de información de contexto del usuario.

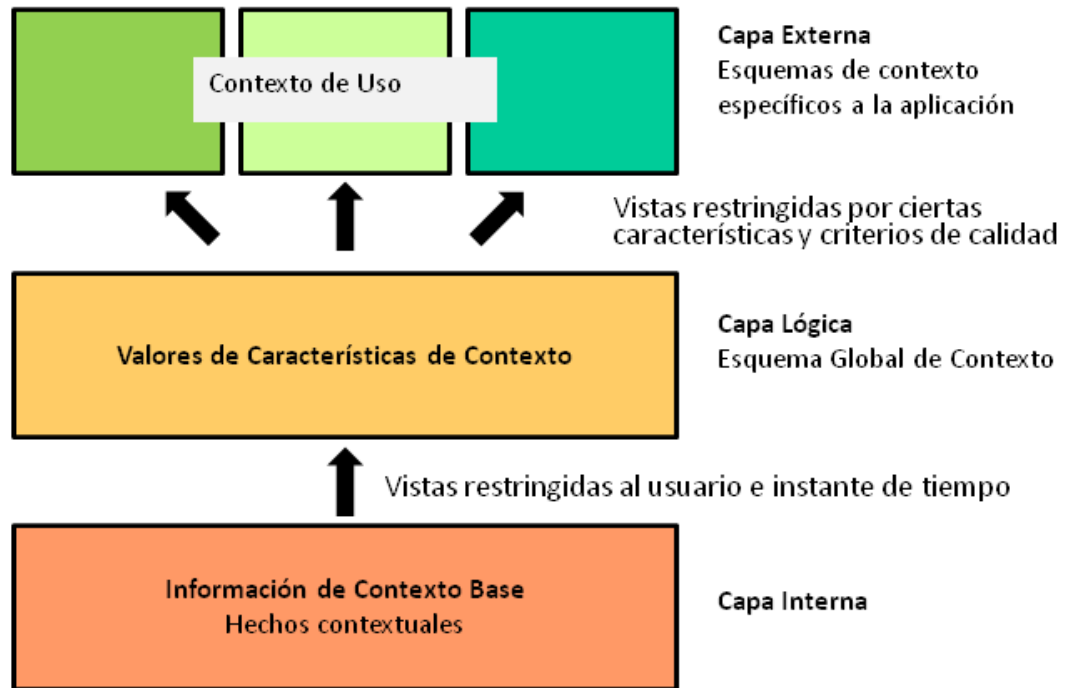


Figura 16. Modelo Contextual por Capas [132]

A continuación se describe cada una de las capas:

- *Capa Interna:* Almacena toda la información recolectada acerca de los usuarios en una forma dependiente del tiempo. Puede contener datos imperfectos, debido a que es proporcionada por sensores ambientales.
- *Capa Lógica:* Proporciona una completa visión del contexto de un usuario en un instante de tiempo específico junto con los metadatos imperfectos que permiten determinar la confiabilidad de la información almacenada. Se eliminan las inconsistencias en los datos recolectados con respecto al instante de tiempo considerado, utilizando estrategias de resolución de conflictos.
- *Capa Externa:* representa el uso del contexto de una aplicación particular en un instante de tiempo determinado. La información contextual está en un esquema que la aplicación entiende y también se garantiza un cierto nivel de criterio de calidad (tal como un mínimo de confianza).

Este esquema de capas asume que la mayoría de las aplicaciones y la generalidad de fuentes de datos contextuales tendrán acceso al sistema a través de la capa externa, utilizando un esquema propio; sin tener que tratar con la complejidad de los datos contextuales y los problemas de gestión asociados. La capa intermedia resolverá los conflictos y calculará la confiabilidad de la información recolectada. Esta capa puede ser mejorada en las capas inferiores

añadiendo heurísticas apropiadas e incluyendo servicios de razonamiento para tratar con información ontológica antecedente.

Inspirados en este modelo nosotros proponemos desarrollar sistemas de bases de datos sensibles al contexto con un modelo de tres capas que se muestra en la figura 17.

En la capa interna se almacenan datos en bruto de los usuarios y una meta-base de conocimiento contextual. Además por esta capa se aceptan datos contextuales relativos del ambiente en que se efectúa la interacción.

En una capa intermedia tendremos módulos de Entrada-Salida que sirvan de mediadores entre la capa externa donde estarán los usuarios con sus definiciones de contexto y la capa interna de datos en bruto. Tanto para el almacenamiento como para la recuperación de los datos los módulos tendrán en cuenta el conocimiento de la meta-base de datos contextual y el contexto en el esquema de interacción del usuario. Aquí el contexto del usuario está integrado con respecto a todos los usuarios.

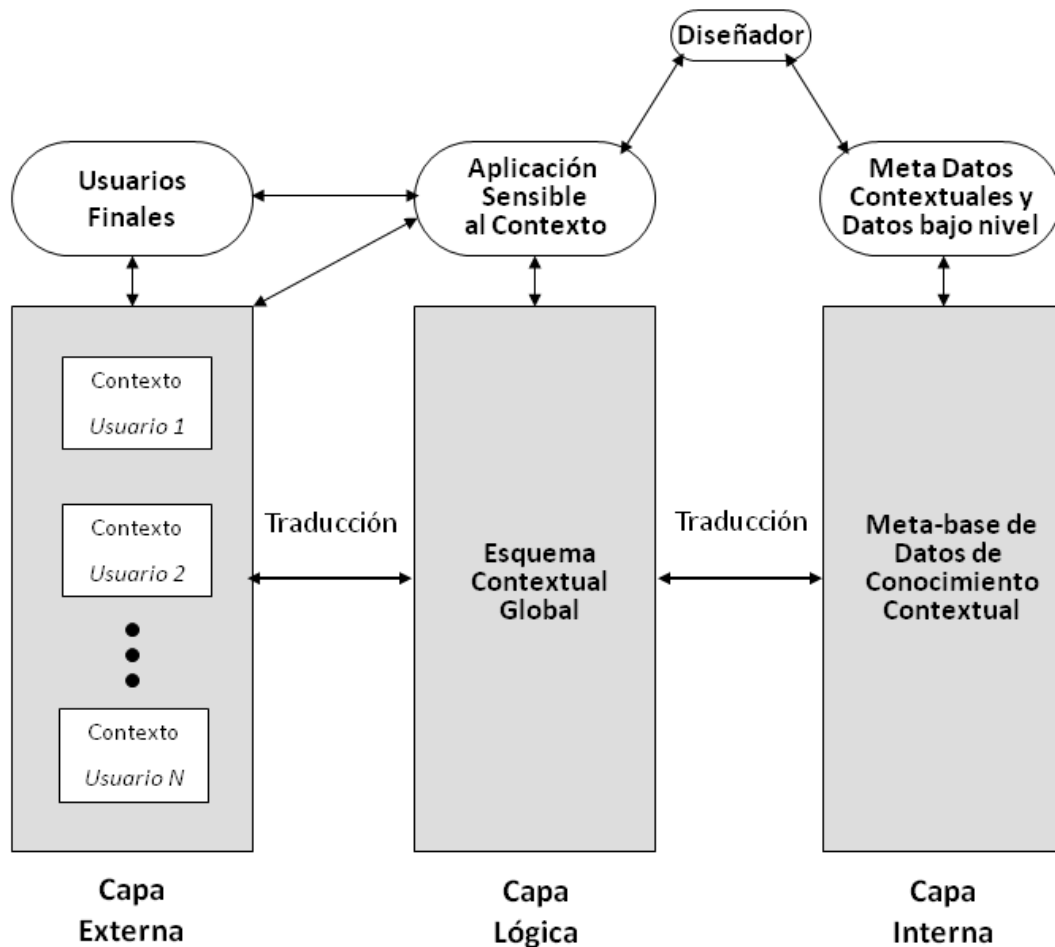


Figura 17. Modelo Contextual por Capas propuesto

En la capa externa interactúan los usuarios y las aplicaciones sensibles al contexto, para introducir y obtener información que ya está sensibilizada por el contexto en el momento de la interacción. Cada usuario tiene su contexto debido a que puede definir aspectos específicos tal como perfil, rol, tarea, además de indicar cuáles de estas dimensiones desean que sensibilicen su interacción.

Es importante destacar que actores principales para que funcionen adecuadamente sistemas como los que estamos proponiendo son tanto los usuarios finales como los diseñadores del sistema, pero ambos con roles muy diferentes. Lo que se plantea es que el usuario y las aplicaciones interactúen en el más alto nivel de abstracción y le sean transparentes problemas relativos a la adquisición de datos contextuales de bajo nivel, además de la definición de meta datos contextuales.

Los usuarios finales tendrán la responsabilidad de definir su esquema contextual personalizado, como veremos, en nuestra prueba de concepto utilizamos un lenguaje estándar como el SQL para que pueda ser utilizado en una interacción de usuarios con cierta experticia que conozcan dicho estándar; pero también proponemos el desarrollo de aplicaciones para usuarios no expertos para que los usuarios interactúen a través de interfaces amigables.

Por otro lado, utilizando el modelo contextual que propondremos el diseñador puede enfocarse en definir metadatos contextuales que deben ser almacenados en una meta-base de conocimiento, de acuerdo al dominio de la aplicación en particular, para poder dotar de la lógica de sensibilización al contexto necesaria tanto al sistema gestor como a las aplicaciones.

4.4.2. Integración del Contexto en un SGBD

Una vez conocidos los requerimientos para la integración de la sensibilización al contexto y estudiado el modelo contextual por capas, podemos analizar cómo sería la integración del contexto en un SGBD para poder lograr nuestro objetivo general de presentar una arquitectura para Sistemas de Bases de Datos Sensibles al Contexto.

En la figura 18 se muestra el esquema general propuesto por Pitoura, Stefanidis y Zaslavsky [116] para integrar el contexto en un Sistema Gestor de Bases de Datos (SGBD).

Los componentes que se muestran en la figura son los siguientes:

- Los “Módulos de Sensibilización” (*Awareness Modules*) se comunican con las fuentes que producen los datos (por ejemplo, sensores) y propagan las actualizaciones al Gestor de Contexto.

- El “Gestor de Contexto” (*Context Manager*) es responsable por administrar (modelar, almacenar, actualizar) la información relacionada al contexto.
- El “Repositorio de Contexto” (*Context Repository*) es el módulo donde el contexto es almacenado.

Hay dos formas de integrar el contexto en un SGBD: a) El gestor de contexto puede ser parte del SGBD o b) El gestor de contexto puede ser visto como un mediador en una capa intermedia.

Como veremos, nuestra propuesta incorpora el gestor como una capa intermedia, pero con la ventaja de utilizar las capacidades que proporciona un Sistema Gestor de Bases de Datos Objeto-Relacional (SGBDOR), con dos propósitos principales:

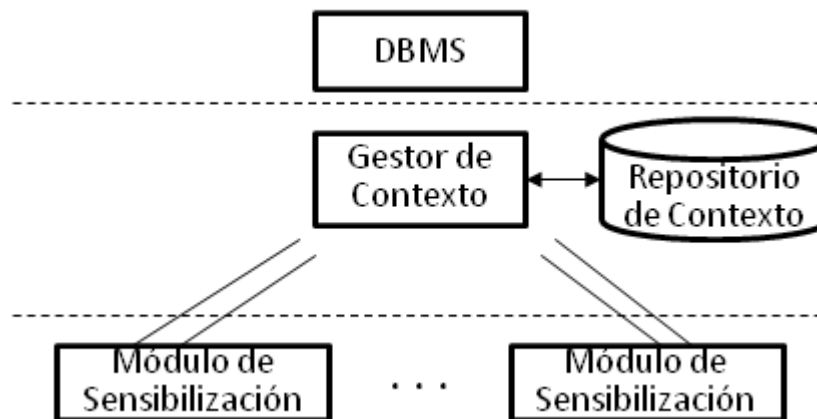


Figura 18. Conectando el Contexto y las Bases De Datos [116]

- 1) Utilizar sentencias estándar, tratando de ser lo menos intrusivo posible en un nuevo lenguaje de interacción que debe ser utilizado por los usuarios y
- 2) Afectar en menor medida (que las propuestas de capa intermedia hechas hasta ahora) el rendimiento del Sistema de Bases de Datos, ofreciendo nuevas funcionalidades que le dan flexibilidad a las consultas de los usuarios.

Tendremos módulos de sensibilización (que, como se presenta en el próximo capítulo, usan resultados de la Teoría Computacional de Percepciones para acercarnos al lenguaje natural como herramienta de interacción) tanto para el almacenamiento como para la recuperación de datos. Estos módulos se comunicarán en la capa externa con los usuarios (expertos) y las aplicaciones

(para usuarios inexpertos), mientras que en la capa interna lo harán con el gestor del contexto.

Por otra parte contaremos con un gestor del contexto que se encargará de consultar los metadatos contextuales y hechos contextuales, que conformarán nuestra meta-base de conocimiento contextual, como el módulo inteligente que permita la sensibilización al contexto.

4.4.3. Nuestra propuesta de arquitectura

Basándonos en la discusión anterior, en la figura 19 se muestra la arquitectura general para el desarrollo de un Sistema de Bases de Datos Sensible al Contexto.

Los componentes de la arquitectura se describen a continuación.

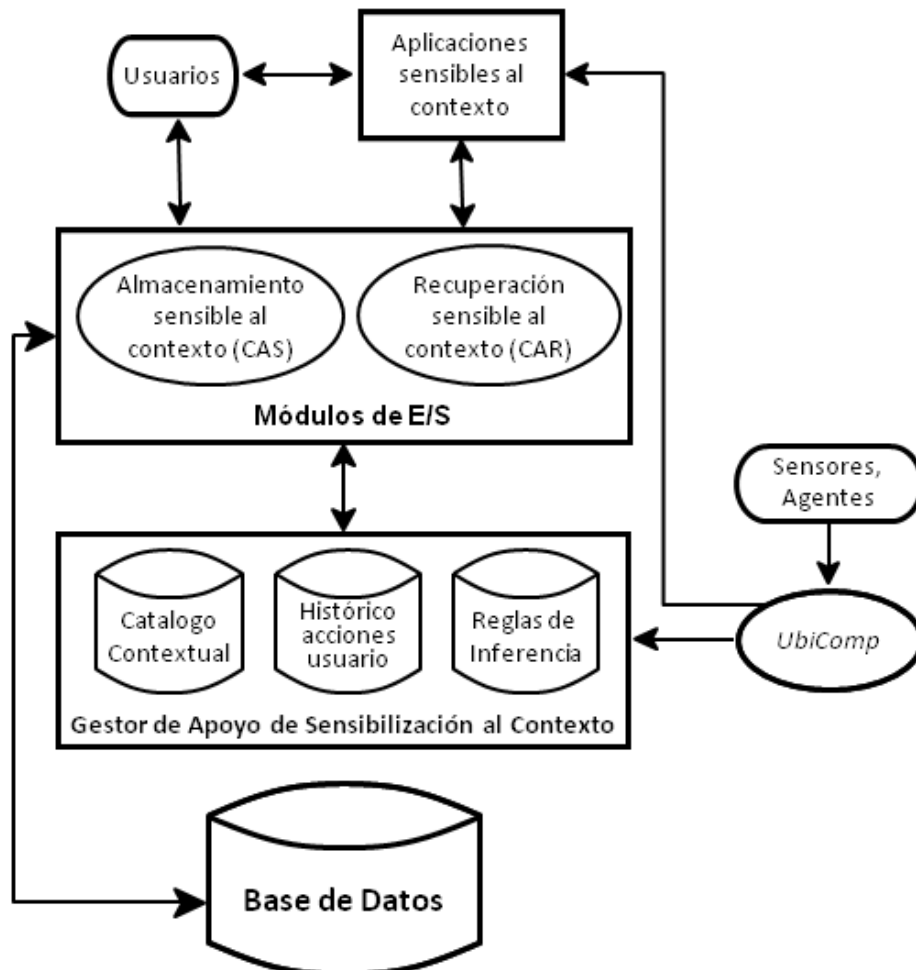


Figura 19. Arquitectura para el desarrollo de un Sistema de Bases de Datos Sensible al Contexto

- *Usuarios*: Los usuarios a través de los módulos de sensibilidad al contexto interactúan con el sistema para aportar instrucciones e información de contexto de manera explícita, como la que recoge su perfil. La interacción puede ser efectuada a través de instrucciones del lenguaje estándar del SGBDOR o a través de aplicaciones sensibles al contexto diseñadas para usuarios inexpertos.
- *Módulos de E/S*: Consta de los módulos de almacenamiento y recuperación sensibles al contexto (*CAS* y *CAR* respectivamente); éstos son los encargados de comunicarse con el *Gestor de apoyo de Sensibilización al Contexto* para que sean adaptadas las entradas (o salidas) de datos de acuerdo al contexto, antes de ser almacenados (o recuperados) los datos en (o de) la base de datos.
- *Gestor de Apoyo de Sensibilización al Contexto (CASM)*: es el encargado de gestionar los datos contextuales en el Catalogo Contextual, el histórico de las acciones de los usuarios y las reglas que permitan inferir comportamiento de acuerdo al contexto. Este sistema se convierte en el *experto* que permite a los *Módulos E/S* hacer su trabajo de transformación.
- *Bases de Datos*: gestiona datos dando soporte a los datos del dominio como a los datos relativos al contexto y sus metadatos.
- *Sensores: Contexto explícito*: permiten aportar información de contexto relativa al ambiente tanto a aplicaciones de usuario como al gestor de apoyo a la sensibilidad al contexto.

4.5. Aportes

Las contribuciones que fueron presentadas en este capítulo se resumen en:

- Definición del esquema de Interacción de un Usuario con el Sistema de Bases de Datos Sensible al Contexto.
- Definición del Modelo Contextual en un esquema de Interacción de un Usuario con el Sistema de Bases de Datos Sensible al Contexto.
- Caracterización del Módulo de Sensibilización al Contexto.
- Arquitectura que sirva como marco de trabajo para el desarrollo de aplicaciones de Bases de Datos Sensibles al Contexto.

Con el modelo propuesto permitimos que las consultas en aplicaciones de Bases de Datos Sensibles al Contexto puedan producir resultados diferentes dependiendo del contexto en el cual son ejecutadas.

En el próximo capítulo nos enfocaremos en una particularización del modelo propuesto para presentar el marco teórico que permita desarrollar atributos lingüísticos con semántica adaptable, que justifica la utilización de Bases de datos Difusas en la capa física.

Capítulo 5

Atributos Lingüísticos con Semántica Adaptable al Contexto

En el capítulo anterior, hemos analizado el esquema de interacción de un usuario con un sistema de bases de datos sensible al contexto y, en dicho marco, hemos definido un modelo de contexto, hemos analizado la necesaria adaptación de la salida del sistema y hemos propuesto una arquitectura de sistema que sirve de referencia para la construcción de SGBDSC. En este capítulo, utilizamos estos conceptos para abordar un caso concreto de sensibilidad al contexto en el ámbito de los sistemas de información: la semántica del lenguaje adaptada al contexto.

Ya hemos comentado brevemente en los capítulos anteriores que la cuestión de la semántica del lenguaje es un elemento fundamental a la hora de establecer sensibilidad al contexto, en aquellos sistemas en los que la interacción se hace a través de un lenguaje más o menos cercano al lenguaje natural. En el ámbito de los sistemas de bases de datos, una de las áreas que se ha preocupado por esa cercanía es el área de la Consulta Flexible de Bases de Datos y dentro de ella, en especial, aquellas propuestas que han incorporado conceptos de la Teoría de Conjuntos Difusos y la Teoría Computacional de Percepciones. Uno de los elementos de dichas teorías, cuya incorporación ha resultado más beneficiosa para facilitar la interacción con el usuario, es la mejora de los lenguajes de consulta mediante el uso de etiquetas lingüísticas junto con los convencionales valores precisos. Las bases de datos, gracias a esta incorporación, han dado un paso firme de mejora en la comunicación con el usuario permitiendo tanto la representación y gestión de información imperfecta, como la consulta *flexible* con términos imprecisos de uso cotidiano por los usuarios. De esta manera es posible preguntarle a una base de datos por *estudiantes jóvenes con buena nota* en lugar de utilizar consultas de intervalos convencionales.

Sin embargo, este paso en firme hacia la comunicación amigable con el usuario, plantea un reto muy relacionado con la sensibilidad al contexto de los sistemas de bases de datos que nos ocupa en esta memoria. Y es que la semántica de los términos lingüísticos utilizados, varía de unos usuarios a otros; es más, siendo el usuario el mismo, varía en relación con otros aspectos del contexto.

Volviendo al ejemplo anterior, la definición de *estudiantes jóvenes con buena nota* no sólo depende de quién la plantea sino de para qué la plantea, por ejemplo.

Por tanto, si cualquier sistema de bases de datos es un candidato adecuado para hacerlo sensible al contexto, las bases de datos difusas son candidatos idóneos donde se plantea un reto si cabe superior. Por desgracia, tal como se evidenció en los antecedentes, las propuestas que se encuentran en la literatura de las bases de datos difusas no han resuelto este problema de manera satisfactoria.

Por este motivo, como otra de las aportaciones de esta tesis, en este capítulo proponemos un modelo de representación y gestión de atributos lingüísticos sensibles al contexto, como un primer paso necesario para la consecución de bases de datos difusas sensibles al contexto.

En las siguientes secciones presentamos en primer lugar el problema de los atributos sensibles al contexto en Bases de Datos Difusas, luego proponemos un Modelo Teórico Formal donde estén incluidas éstas y finalmente presentamos su implementación transparente en un Sistema Gestor de Bases de Datos Objeto Relacional (SGBDOR).

5.1. El problema de los atributos sensibles al contexto en Bases de Datos Difusas

Ya hemos visto en el capítulo 3 las ventajas y capacidades de las bases de datos difusas. Este tipo de bases de datos aportan herramientas variadas para la representación y consulta de información imperfecta. En este capítulo, nos vamos a centrar en una de esas herramientas: los atributos lingüísticos. De manera sencilla, podemos considerar que un atributo lingüístico es un atributo que cuenta dentro de su dominio con términos lingüísticos. Existen muchos tipos de atributos en la literatura que gozan de esta cualidad, si bien, en esta memoria, nos vamos a centrar en aquellos que en la bases de datos tienen una semántica subyacente y la expresan mediante herramientas de la Teoría de Conjuntos Difusos. En esta memoria, los denominaremos dominios difusos.

Las etiquetas lingüísticas empleadas por el usuario en relación con datos de estos dominios para comunicarse con el sistema, pueden tener diferente semántica de acuerdo con el contexto de interacción. Por ejemplo en el área médica podríamos tener variables lingüísticas como el peso, la altura, o flexión de la cadera; en el caso de la altura puede ser descrita con las etiquetas lingüísticas *bajo*, *normal*, *alto*, las cuales podrían interpretarse de manera diferente con respecto al usuario y la actividad que el usuario está llevando a cabo (por ejemplo, exploración general, la exploración quirúrgica previa o examen post-quirúrgico).

En la próxima sección mostramos la base teórica para la aplicación de la semántica adaptable de etiquetas lingüísticas utilizando dominios difusos.

5.1.1. Dominios Difusos

Como hemos comentado, una base de datos difusa se apoya en la gestión de dominios difusos para modelar y representar datos y flexibilizar las consultas. En la figura 20 se presenta una jerarquía de clases que incluye los dominios difusos diseñados en el modelo propuesto por Cuevas [41] descrito con detalle en el apartado 3.15 y con representación gráfica en la figura 7, utilizando las características de un sistema objeto-relacional.

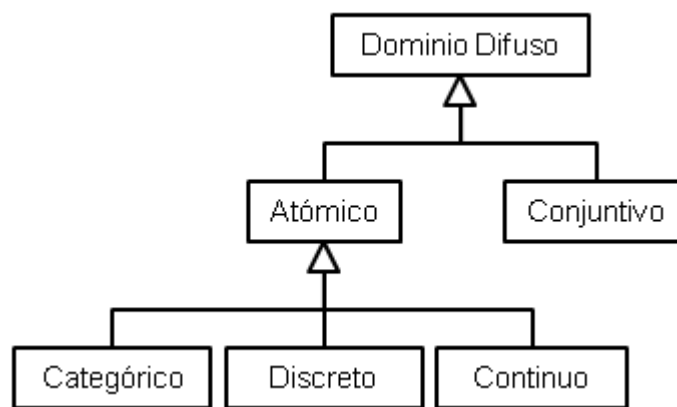


Figura 20. Jerarquía de Dominios Difusos

Partiendo de lo que se indicó en dicho apartado, hemos efectuado la siguiente categorización. Se pueden distinguir dos tipos de dominios difusos, de acuerdo a la cardinalidad, a saber:

- *Atómico*, son aquellos en los cuales el atributo puede tomar un solo valor que puede ser difuso, a su vez se pueden dividir en:
 - *Categórico*: las etiquetas lingüísticas no tienen representación semántica asociada y se crea una matriz de semejanza entre estas etiquetas.
 - *Discreto*: las etiquetas lingüísticas son expresadas mediante una definición extensiva de un conjunto difuso donde el conjunto soporte tiene una escala discreta.
 - *Continuo*: las etiquetas lingüísticas son definidas por medio de funciones (normalmente lineales) donde el conjunto soporte es continuo (por ej. números reales). En la práctica, generalmente se utilizan funciones (unimodales o monótonas) utilizando varias

rectas en su representación gráfica, constituyendo la forma de un trapecio (completo o incompleto) o un triángulo. Estas figuras se representan normalmente por cuatro puntos: A y D (representan el soporte de la función de pertenencia); B y C (representan el núcleo de la función de pertenencia). Debido a la facilidad para su manipulación matemática, el uso de distribuciones trapezoidales como forma de generalizar el resto de funciones es ampliamente aceptado en la comunidad de investigadores de bases de datos.

- *Conjuntivo*: permiten que el atributo tome como valor un conjunto difuso que representa un valor múltiple.

Este modelo puede ser usado por diseñadores de aplicaciones de bases de datos, para definir la estructura y el esquema de la base de datos a través de un SGBDOR, o crear interfaces que permitan a un usuario inexperto interactuar con el sistema o para expresar sus percepciones tanto a nivel de definición de etiquetas lingüísticas, como en la obtención de información flexible. Una implementación de prueba de concepto puede encontrarse en [42].

A continuación se muestran ejemplos de los tipos de dominio difuso en el área médica, específicamente en el ámbito del problema hospitalario que desarrollamos en los próximos capítulos.

Ejemplo 11. *Dominio Difuso Atómico Categórico.*

En el laboratorio de marcha del Hospital Ortopédico Infantil de Venezuela el paciente camina a lo largo de una pista de 10 metros y el evaluador, al observar las características más relevantes, coloca una etiqueta para el *tipo de marcha* que puede ser: normal, hemipléjica, atáxica, parkinsoniana, danzante, antiálgica o espástica. Dado que no puede asociarle a cada etiqueta un valor semántico, lo que se hace es llenar una matriz de semejanza donde se coloca un grado de pertenencia entre cada par de tipo de marcha. Por ejemplo se puede colocar un grado de parecido de 0.7 entre una marcha normal y una antiálgica.

Ejemplo 12. *Dominio Difuso Atómico Discreto.*

El tono muscular es la capacidad de contracción de un músculo; se mide en una escala de 1 a 4, siendo (1) Falta de tono y (4) Exceso de tono con espasticidad. Pero también el tono muscular puede ser representado por etiquetas tales como: atonía, hipotonía, normotonía e hipertonia. En este caso se establece una matriz de pertenencia, pero entre cada par formado por un número de la escala de medición y una etiqueta lingüística. Por ejemplo, se puede colocar un grado de pertenencia de 0.8 entre 1 e hipotonía. Como ejemplo de un atributo en el dominio de la aplicación se tiene el tono flexor dorsal (derecho e izquierdo), el cual mide la capacidad de contracción del músculo del tobillo al acercar el pie derecho al plano

longitudinal del tobillo con la rodilla a 90 grados de flexión y es medido en una escala de 0 al 5.

Ejemplo 13. *Dominio Difuso Atómico Continuo.*

El rango articular son los límites de grado en que el paciente logra realizar el movimiento con cualquier articulación. Es por ello que en un atributo como la flexión de la cadera se miden los grados cuando el miembro inferior se acerca al tronco anterior en el plano longitudinal. Es claro que este tipo de atributo tiene un universo de discurso cuyo dominio es continuo (0 a 360 grados), pero usualmente los fisioterapeutas por observación colocan etiquetas como “*Dentro de los límites normales o DLN*”, *alto* o *bajo*, para describir la variable lingüística.

Ejemplo 14. *Dominio Difuso Conjuntivo*

En nuestro un paciente puede utilizar dispositivos para mejorar la marcha indicados por los médicos referentes; tales como: andaderas, bastón, muletas, férulas, AFO, etc. Un paciente puede utilizar más de un dispositivo, por ejemplo bastón y andaderas; en este caso se puede utilizar un conjunto difuso para indicar con qué frecuencia utiliza cada uno de los dispositivos, así como {bastón/0.8; Andadera/0.6}

5.1.2. El problema de la Semántica

Si volvemos a la arquitectura presentada en el capítulo anterior, en este caso se construirá la arquitectura alrededor de una Base de Datos Difusa en la capa del SGBD. Más adelante en este capítulo, comentaremos cómo se desarrolla dicha arquitectura para implementar un sistema con atributos de semántica adaptable al contexto.

En la figura 21, se muestra el modelo de sensibilización al contexto utilizando una base de datos difusa. En este caso, los módulos de mediación con el usuario utilizan conceptos de la Teoría de Conjuntos Difusos y la Teoría computacional de percepciones para *entender* los términos que utiliza el usuario para describir los datos y emplearlos en consecuencia.

Toda la lógica que se le añada al sistema gestor de bases de datos difusas debe ir encaminada a que la interpretación de los términos lingüísticos empleados por el usuario sea sensible al contexto.

Por ejemplo un médico puede definir que “DLN” para la flexión de la cadera está entre 120° y 130°. Esta etiqueta puede ser descrita por los cuatro puntos de inflexión como la 4-tupla: (110, 120, 130, 140). Pero esta definición puede cambiar para otro médico que puede considerar una forma triangular descrita como: (115, 125, 125, 135).

Asimismo, debido a que el tipo de marcha se describe por etiquetas categóricas y la matriz de semejanza es determinada por cada usuario, el médico 1 puede asignar un grado de 0.7 entre marcha normal y antiálgica, mientras que para el médico 2 puede ser un grado de 0.8. Lo mismo aplica para definir las funciones de pertenencia en el caso del tono flexor.

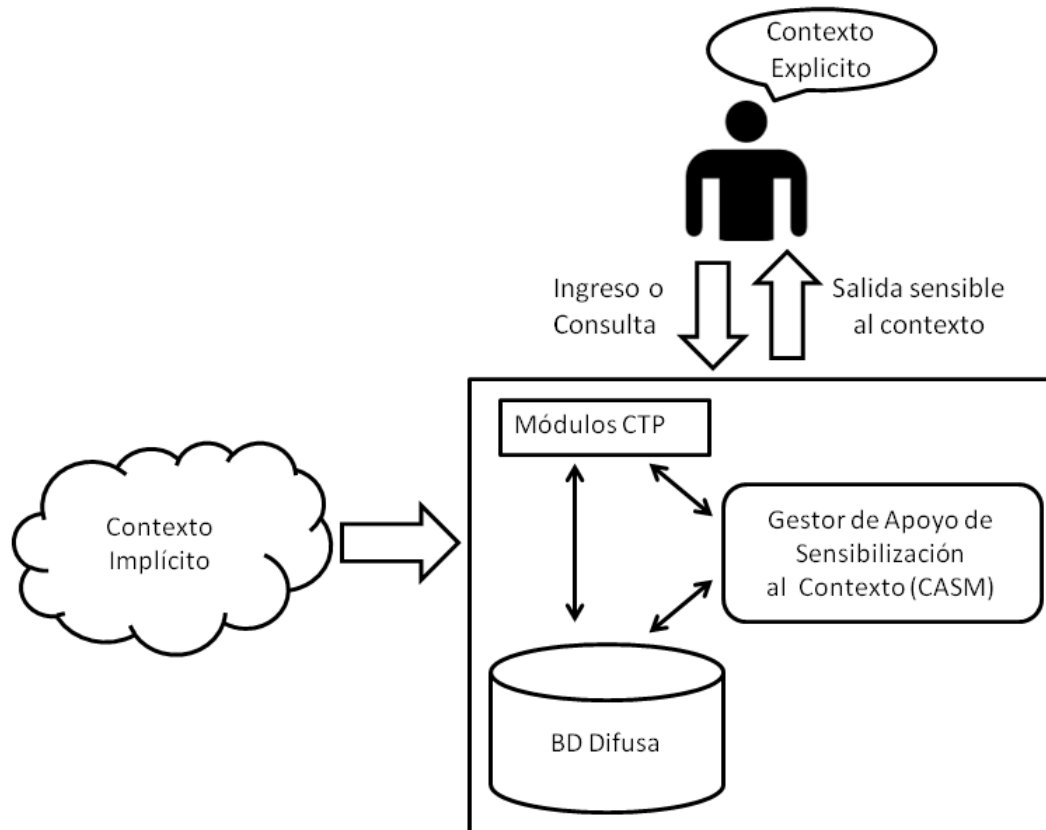


Figura 21. Modelo Sensibilización al Contexto en Sistemas de Información

Ahora bien, el usuario1 también puede considerar la tarea (examen preoperatorio, general o postoperatorio) para definir diferentes significados de la etiqueta DLN. Por ejemplo, DLN cuando se le hace un examen post-operatorio a un usuario puede estar representado por la 4-tupla: (100, 110, 120, 130).

5.2. Modelo Formal Propuesto

Para permitir la utilización de atributos lingüísticos con semántica adaptada al contexto en nuestro sistema de bases de datos difuso, necesitamos establecer un modelo formal que permita la representación de la dependencia al contexto de los dominios difusos y manipule los valores de dichos atributos en relación a las dependencias. En nuestro modelo, la dependencia al contexto se gestiona a través del concepto de dependencia contextual y dimensiones

contextuales. Primero establecemos las definiciones pertinentes para explicar luego el modelo de dependencias contextuales.

Definición 4. *Dimensión Contextual.*

Se denomina *dimensión contextual* a cada uno de los aspectos de un modelo de contexto a los que puede mostrarse sensible un sistema de bases de datos. Cuando sea necesario, para un modelo de contexto C , denotaremos por $Dim^C = \{d_1, \dots, d_m\}$ el conjunto de dichas dimensiones.

Es decir, las dimensiones contextuales son cada una de las características o rasgos sobre los que se tiene información que define el contexto de la interacción entre el usuario y el sistema.

Volvamos al modelo de contexto definido en el capítulo 4 (*definición 2*). Según dicho modelo, el contexto es una 3-tupla $\langle u, a, o \rangle$, donde $u \in P \times T \times E \times R$, $a \in TP \times D \times U \times A$ y $o \in O_1 \times \dots \times O_n$, siendo P el conjunto de perfiles, T el conjunto de tareas, E el conjunto de estados, R el conjunto de roles, TP el conjunto de instantes de tiempo, D el conjunto de dispositivos, U el conjunto de ubicaciones, y $O_1..O_n$ otros aspectos del contexto adicionales propios del dominio de aplicación.

En este caso, cada uno de estos conjuntos $P, T, E, R, TP, D, U, A, O_1, \dots, O_n$ representa el dominio de las distintas *dimensiones contextuales* consideradas en nuestro modelo de contexto.

Por ejemplo, en un caso de un dominio de aplicación particular se pueden definir las siguientes dimensiones contextuales:

$$Dim^C = \{perfil, rol, tarea, dispositivo\}$$

Si queremos que la semántica de los valores de un dominio sea sensible al contexto, debemos establecer relación entre dichos dominios y las dimensiones.

Definición 5. *Dominio no dependiente del contexto*

Sean

D un dominio lingüístico definido en una base de datos,

$L_D = \{l_1, \dots, l_n\}$ el conjunto de etiquetas lingüísticas de dicho dominio.

Se dice que D es un dominio no dependiente del contexto si la semántica asociada a cada etiqueta lingüística l_i no varía en relación al contexto.

Si denotamos por $\mu_{l_i}^{v_C}$ a la función de pertenencia que define la semántica de la etiqueta l_i sobre un referencial R en un contexto v_C determinado (instancia del modelo de contexto C), en un dominio lingüístico no dependiente del contexto se cumple la siguiente propiedad para cualesquiera dos instancias del modelo de contexto C :

$$\forall x \in R, \forall v_C, v'_C, \mu_{l_i}^{v_C}(x) = \mu_{l_i}^{v'_C}(x)$$

Definición 6. *Dominio dependiente del contexto*

Sean

D un dominio lingüístico definido en una base de datos,

$L_D = \{l_1, \dots, l_n\}$ el conjunto de etiquetas lingüísticas de dicho dominio.

Se dice que D es un dominio dependiente del contexto si la semántica asociada a cada etiqueta lingüística l_i puede variar en relación al contexto.

Es decir, un dominio dependiente del contexto no está sujeto a la anterior restricción.

Definición 7. *Conjunto de dependencias de un dominio.*

Sea D un dominio lingüístico definido en una base de datos con sensibilidad a un modelo de contexto C caracterizado por el conjunto de dimensiones $Dim^C = \{d_1, \dots, d_m\}$. Llamaremos conjunto de dependencias de D y lo denotaremos por $Dset(D)$ al conjunto de dimensiones de Dim^C a las que es sensible D .

De la definición anterior se sigue que:

$$D \text{ es no dependiente del contexto} \leftrightarrow Dset(D) = \emptyset$$

Definición 8. *Atributo dependiente del contexto.*

Sean

$T = \{A_1, \dots, A_m\}$ una relación de una base de datos,

D_1, \dots, D_m los dominios de los atributos A_1, \dots, A_m , respectivamente.

El atributo A_i es un atributo dependiente del contexto si su dominio D_i es un dominio dependiente del contexto.

Definición 9. *Conjunto de dependencias de un atributo.*

Sean

$T = \{A_1, \dots, A_m\}$ una relación de una base de datos.

D_1, \dots, D_m los dominios de los atributos A_1, \dots, A_m , respectivamente.

A_i un atributo dependiente del contexto según un modelo C definido por el conjunto de dimensiones $Dim^C = \{d_1, \dots, d_n\}$.

Llamaremos conjunto de dependencias de A_i y lo denotaremos por $Dset(A_i)$ al conjunto de dimensiones de Dim^C a las que es sensible D_i . Es decir,

$$Dset(A_i) = Dset(D_i)$$

Definición 10. *Conjunto de dependencias de una relación.*

Sean

$T=\{A_1, \dots, A_m\}$ una relación de una base de datos.

D_1, \dots, D_m los dominios de los atributos A_1, \dots, A_m , respectivamente.

Llamaremos conjunto de dependencias de T y lo denotaremos por $Dset(T)$ al conjunto de dimensiones de Dim^C a las que son sensibles los atributos de T :

$$Dset(T) = \bigcup_{i=1..n} Dset(A_i)$$

Definición 11. *Relación dependiente del contexto.*

Sea

$T=\{A_1, \dots, A_m\}$ una relación de una base de datos.

Se dice que T es una relación dependiente del contexto si y sólo si:

$$Dset(T) \neq \emptyset$$

La aplicación de esta definición en la práctica permite representar en el lenguaje del usuario los datos almacenados en bruto (objetos trapecio) a través de un conjunto difuso, convirtiendo así la semántica en adaptable a las preferencias del usuario. Éste es una de las contribuciones importante y novedosa de nuestro trabajo.

Una vez hechas las definiciones y estudiado el Modelo de contexto en el capítulo anterior, en la figura 22 se muestra el modelo de dependencias contextuales. Mediante este modelo se pueden gestionar múltiples dimensiones contextuales y las dependencias contextuales de los usuarios.

Los atributos de una tabla de usuario cuyo dominio sea difuso en la aplicación pueden ser independientes o dependientes del contexto. En el caso de atributos que sean dependientes del contexto el diseñador de la tabla debe indicar el conjunto de dimensiones contextuales que deben tenerse en cuenta cuando se interactúa con el valor del atributo.

A su vez cada dimensión contextual puede ser instanciada por múltiples dominios de dicha dimensión. En nuestro esquema de interacción definido estos dominios están atados a un usuario en particular.

Una vez que el usuario defina las dependencias contextuales, el sistema podrá automáticamente cambiar el lenguaje utilizado en la comunicación con el usuario tomando en cuenta el contexto.

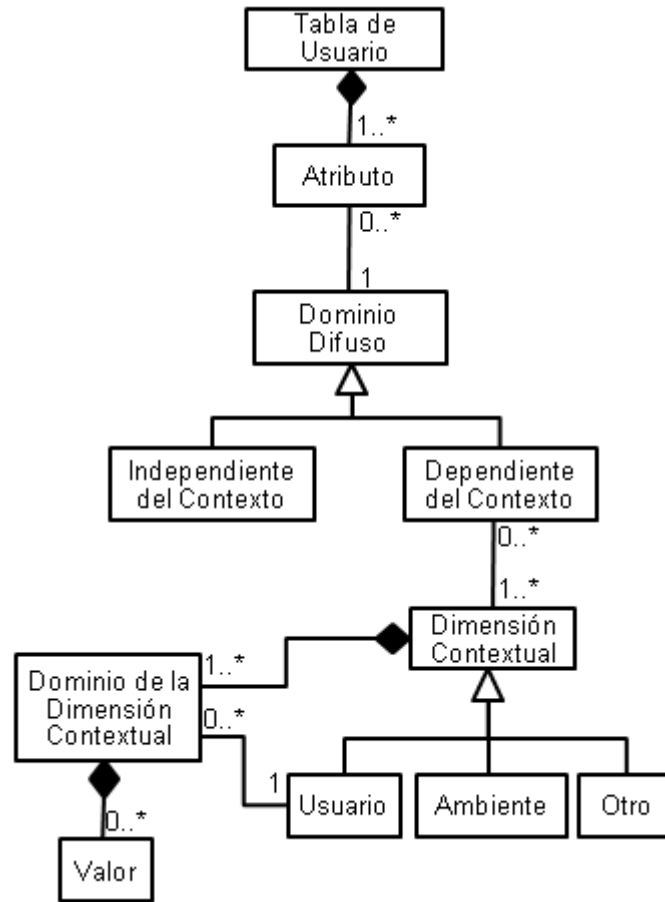


Figura 22. Modelo de Dependencias Contextuales

5.3. Implementación transparente en un SGBDOR

En nuestra propuesta, el conjunto de dimensiones contextuales conforma un catálogo de dimensiones contextuales personalizable; este catálogo es elaborado cuando se está diseñando la aplicación de acuerdo con el dominio específico.

Cada dimensión contextual incorpora un dominio de valores, de modo que el sistema puede ser consciente al valor concreto de las dimensiones en el contexto de interacción. El diseñador del sistema puede definir dominios de referencia por defecto para las dimensiones contextuales, aunque cualquier usuario también puede incluir dominios particularizados en su perfil de usuario.

Las dimensiones contextuales y sus correspondientes dominios se almacenan en el Gestor de Apoyo a Sensibilización del Contexto (CASM), formando parte de la de meta-base de conocimiento contextual.

Una vez definido el catálogo de dimensiones contextuales, el usuario podrá establecer la dependencia contextual apropiada mediante la indicación de las dimensiones contextuales relevantes para cada dominio de atributo en sus tablas.

Posteriormente, los valores de los atributos con estos dominios se comportarán con sensibilidad al contexto: cuando un usuario se conecta para efectuar la inserción de datos o la recuperación de información, el sistema busca el contexto actual, establece una correspondencia entre dicho contexto y la lógica de sensibilización almacenada en el catálogo y procede a realizar la operación requerida en los datos.

Basada en la arquitectura mostrada en el Capítulo 4 (figura 19) se propone una arquitectura para Sistemas de Bases de Datos Difusas Sensibles al contexto (figura 23).

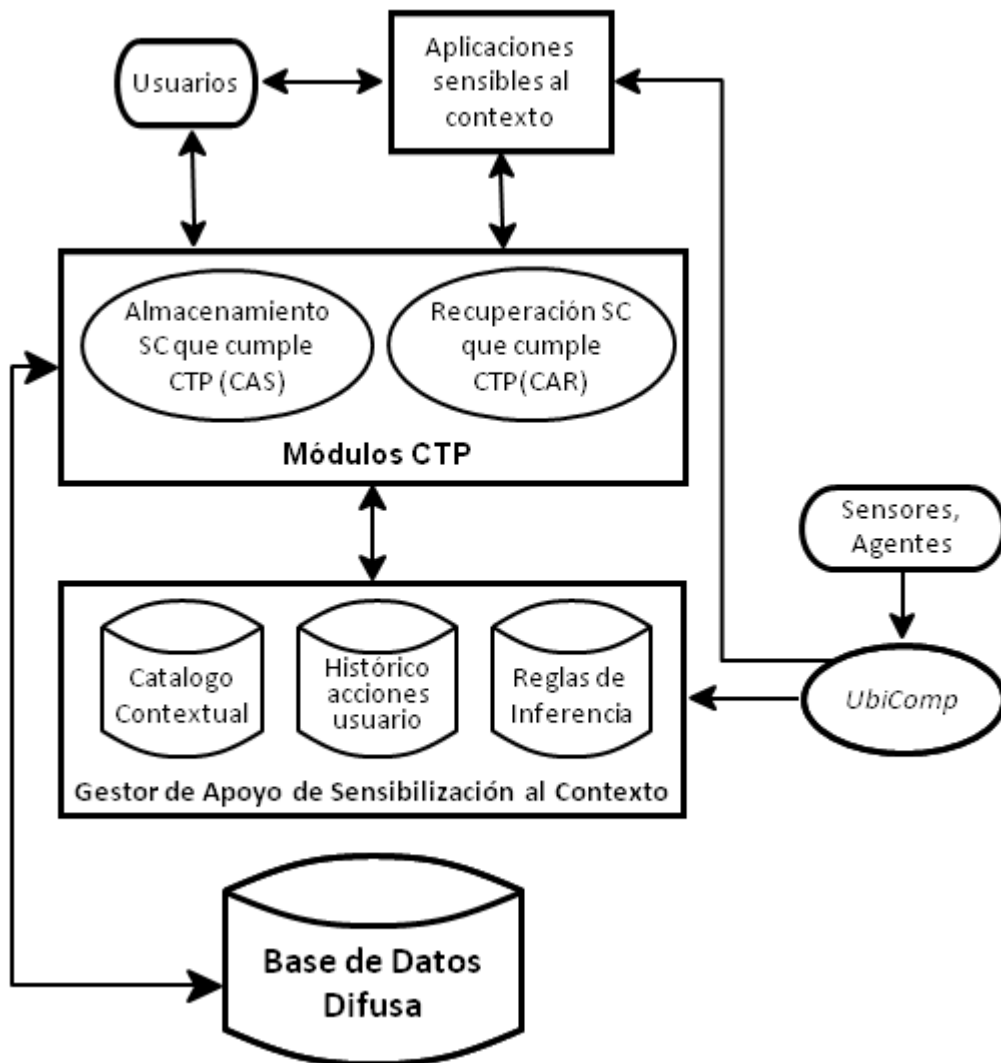


Figura 23. Arquitectura para Sistemas de Bases de Datos Difusas Sensibles al Contexto

La innovación introducida en nuestra propuesta es la utilización en la capa física de una Base de Datos Difusa, lo que permite a los Módulos de Entrada/Salida almacenar datos imperfectos y recuperar información en forma flexible. Dado que tienen esta característica, a estos módulos los denominamos módulos que cumplen de Sensibilización al Contexto (SC) que cumplen con la Teoría Computacional de Percepciones (CTP) o simplemente *Módulos CTP*.

5.3.1. Gestor de Apoyo a la Sensibilización al Contexto

Es obvio que en un Sistema Sensible al Contexto se necesita un módulo que gestione (modele, almacene, actualice) la información relacionada al contexto. En nuestra propuesta este Gestor lo conforma el Catalogo Contextual, histórico de acciones del usuario y reglas de inferencia; éstas últimas son establecidas a través de funciones o procedimientos del lenguaje estándar del SGBDOR y mediante el diseño de métodos de tipos de datos extendidos (dominios difusos).

En la Figura 24, se presenta un esquema del funcionamiento del este Módulo. Pueden utilizarse otros agentes de software para extraer o inferir datos contextuales de acuerdo a las acciones que tome el usuario (uso de aplicaciones web, visualización en internet), así como para almacenar en la base de datos un histórico o registro de estas acciones que permita establecer reglas de inferencia de aspectos contextuales. Así mismo, de acuerdo a lo que se estableció en el estudio preliminar, hay diversas investigaciones que infieren preferencias contextuales, experiencias que pueden servir para abordar este problema.

Por otro lado en el catálogo contextual se almacenan datos del contexto que pueden ser objetos de diferentes tipos de dominios difusos que pueden ser definidos en tiempo de diseño. Nuestra propuesta es utilizar la capacidad objeto relacional de gestores de bases de datos para, mediante la extensión de los tipos nativos de datos, gestionar los dominios difusos. Además, con el uso del lenguaje nativo de procedimientos y funciones de SQL lograr un bajo impacto en el rendimiento del sistema; así como afectar, en lo mínimo posible, el tipo de sentencias estándar utilizado por el usuario en su interacción con el sistema, mediante aplicaciones amigables y donde la sensibilización al contexto sea lo más transparente posible.

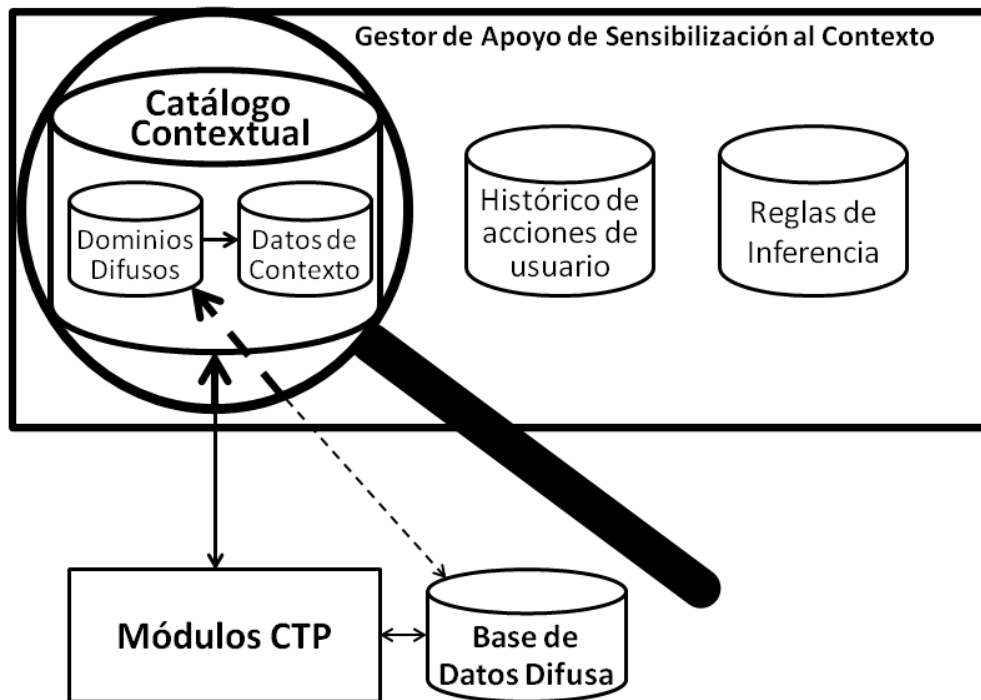


Figura 24. Gestor de Apoyo a la Sensibilización al Contexto

5.3.2. Módulos CTP

A continuación se describe el funcionamiento de cada uno de los módulos; tanto el de almacenamiento de datos (CAS) como el de recuperación de información (CAR) desarrollados en base a la Teoría Computacional de Percepciones (TCP).

5.3.2.1. Módulo de Almacenamiento Sensible al Contexto (CAS)

Debido a que diseñamos mecanismos para la representación de datos imperfectos además de la obtención de consultas difusas aplicando transformaciones lingüísticas de acuerdo al contexto actual del usuario, vamos a mostrar cómo más de un usuario puede hacer uso de sentencias estándar SQL para interactuar con nuestra propuesta de Bases de Datos Difusa Sensible al Contexto (BDDSC).

En la figura 25 se muestra el esquema de interacción con el Sistema cuando intervienen dos usuarios. Cada usuario puede definir su contexto, dimensiones y dominios contextuales, dependencias, además de la semántica de etiquetas según su preferencia. También pueden ser tomados datos contextuales del ambiente en el momento de la interacción de cada usuario. Los módulos CTP

apoyados por el Gestor de Sensibilización realizan las transformaciones lingüísticas pertinentes cuando se almacenan o recuperan datos en bruto.

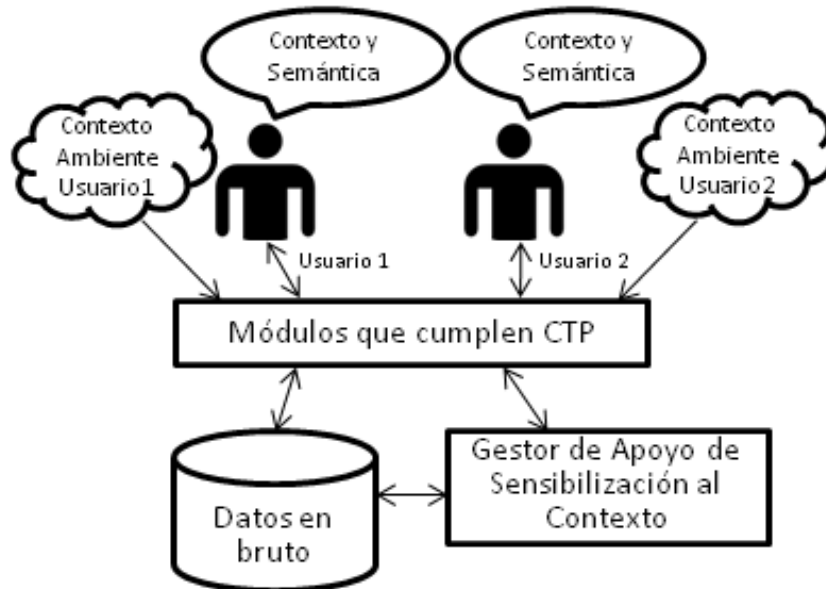


Figura 25. Esquema de Interacción dos usuarios en una BDDSC

En la Figura 26 se presenta un diagrama de flujo que describe el algoritmo seguido por el Módulo de Almacenamiento Sensible al Contexto (CAS) que permite la inserción de datos de forma amigable a un usuario, mediante la implementación de métodos constructores para un dominio difuso atómico continuo definido en una base de datos y la extensión de tipos de datos nativos de un SGBDOR tal como Oracle. A continuación la descripción de las variables, relaciones y métodos involucrados en el algoritmo.

R_Tab : Tabla de la aplicación de n atributos cuyo esquema es $R(A_1, \dots, A_n)$.

D_i : Dominio Difuso correspondiente al atributo A_i , tal que $1 \leq i \leq n$.

$valor$: valor ingresado por el usuario en el atributo A_i en la sentencia INSERT

$MC(Et, tr)$: método constructor del dominio difuso D_i , donde:

Et : etiqueta lingüística, definida por el usuario o por el usuario por defecto.

tr : trapecio descrito por los cuatro puntos de inflexión.

$MC(tr)$: método constructor del dominio difuso D_i , donde:

tr : trapecio descrito por los cuatro puntos de inflexión.

$MC(C)$: método constructor del dominio difuso D_i , donde:

C : valor preciso en el dominio natural del atributo A_i

$OT(tr)$: Objeto tipo trapecio descrito por tr .

$OT(C)$: Objeto tipo trapecio correspondiente al valor preciso C .

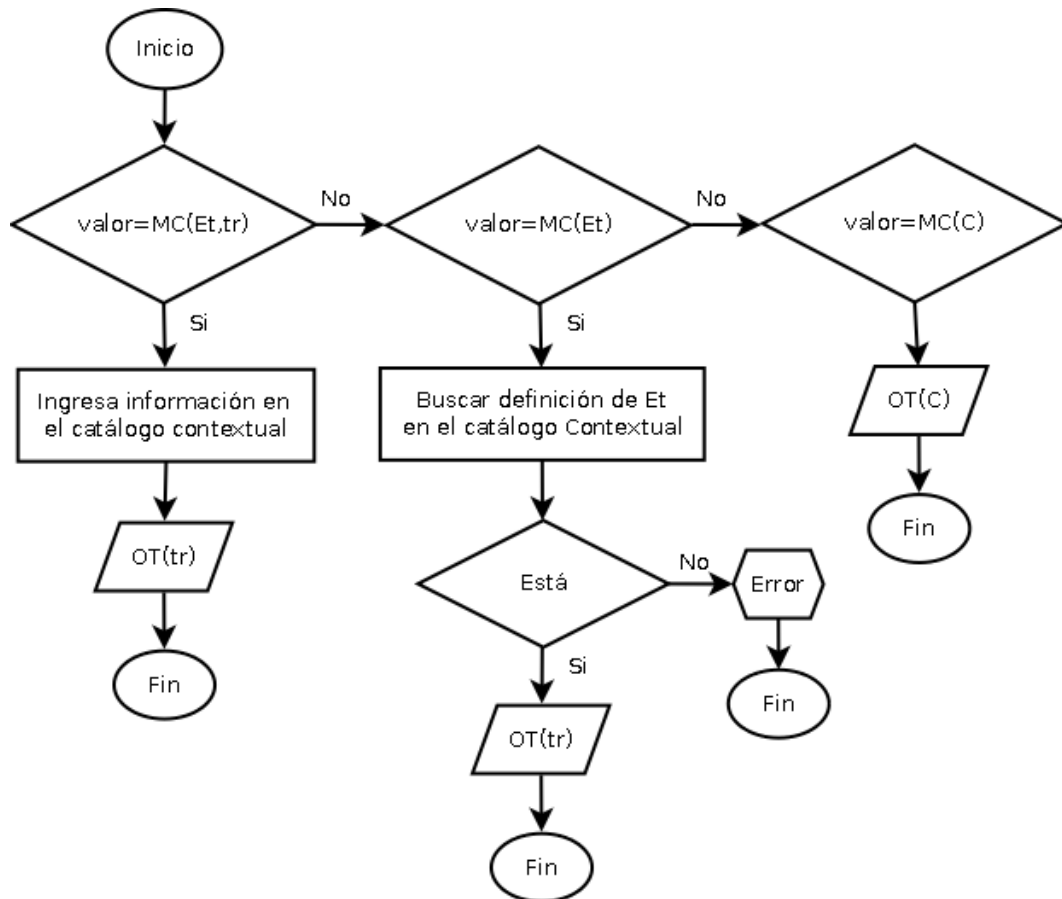


Figura 26. Algoritmo de Inserción implementado por el CAS

Es de hacer notar que el usuario está ingresando datos con una sentencia estándar INSERT, por lo que cada método constructor se activa para ingresar una “traducción” de acuerdo a lo definido en el Gestor de Sensibilización al Contexto (CASM).

Es necesario acotar que el algoritmo ilustra la traducción del CAS para el dominio difuso d_i , pero antes de hacer los INSERT de los datos finales, se hace la traducción de acuerdo a cada método constructor de todos los k valores de v_i .

A continuación se describe la forma general de las sentencias SQL que puede utilizar el diseñador del sistema o un usuario experto en el módulo CAS.

La sentencia CREATE TABLE en una BDDSC tiene la siguiente forma general:

```
CREATE TABLE User_Table (A1 TipoExtendido Restricción_Col,...,
An TipoExtendido Restricción_Col, Retricciones_Tabla)
```

donde,

A1, ... An son atributos

TipoExtendido tipo de dato estándar (carácter, numérico, fecha, ...) o un dominio difuso extendido para aquellos atributos que correspondan a una variable lingüística.

Una de las ventajas de nuestra propuesta es que, gracias a las características objeto-relacionales del SGBD, se puede utilizar un dominio difuso extendido como un tipo de dato nativo. A continuación un ejemplo de la utilización de esta sentencia en el dominio de aplicación que desarrollaremos con más detalle en el próximo capítulo.

Ejemplo 15. Utilización de la sentencia CREATE en una BDDSC.

```
CREATE TABLE Exam_Tab (
    id          NUMBER(10),
    FlexCadDer D_FlexCadera,
    Fecha      DATE,
    profesion  VARCHAR(50),
    marcha    VARCHAR(50),
    CONSTRAINT PK_Ex PRIMARY KEY (id)
);
```

En este caso el usuario está utilizando un tipo extendido (*D_FlexCadera*) para una variable lingüística como la Flexión de la Cadera (rango articular), extendiendo el dominio difuso atómico continuo (definido en la jerarquía de dominios difusos). Este tipo extendido puede ser usado en la tabla Examen (*Exam_Tab*) creada por el diseñador donde se almacenarán valores de un examen físico articular, como tipo de dato del atributo “Flexión Cadera Derecha” (*FlexCadDer*).

Después de creado el tipo extendido y asignado a los atributos correspondientes, el usuario podrá utilizar métodos constructores en las sentencias INSERT que le permitan ingresar valores que representan una función de pertenencia trapezoidal, descrito por etiquetas lingüísticas. La semántica de estas etiquetas es definida por el usuario y puede ser dependiente del contexto, ésta información es almacenada dentro del catálogo contextual del CASM.

Para la prueba de concepto definimos un procedimiento almacenado para que el usuario pueda establecer las dependencias contextuales para los dominios difusos según su interés, la forma general se describe a continuación.

El procedimiento almacenado Agregar Dependencia del Contexto en una BDDSC tiene la siguiente forma general:

```
AgregarDependenciaContexto (DominioDifuso, DimensionContextual)
```

donde,

DominioDifuso es un tipo extendido definido por el usuario.

DimensionContextual es una dimensión contextual.

De acuerdo a nuestra definición de contexto, las dimensiones contextuales pueden ser cualquiera de las definidas en nuestro modelo de contexto (usuario, perfil, tarea, rol, tiempo, dispositivo, ubicación) o cualquier otra que se considere necesario el usuario de acuerdo al dominio de la aplicación. Estas dependencias son almacenadas en el catálogo contextual del CASM.

Ejemplo 16. Dependencias contextuales para el dominio difuso Flexión de la Cadera.

Se pueden establecer dos dependencias contextuales para el dominio difuso Flexión de la Cadera (*D_FlexCadera*) de la siguiente forma:

```
AgregarDependenciaContextual ('D_FlexCadera', 'Usuario')
AgregarDependenciaContextual ('D_FlexCadera', 'Tarea')
```

De esta forma la BDDSC es consciente que los valores del dominio difuso *D_FlexCadera* son sensibles a las dimensiones contextuales *usuario* y *tarea*. Se prevé que puedan definirse dominios contextuales por defecto para cada dimensión contextual. También cada usuario puede proporcionar los dominios que crea necesario. Por ejemplo, en un dominio de aplicación de un hospital, los dominios contextuales de *tarea* pueden ser *examen general*, *pre-operatorio* y *post-operatorio*.

Luego de haber definido las dependencias contextuales, el usuario puede interactuar con la BDDSC utilizando etiquetas lingüísticas y el sistema tomará en cuenta, en forma transparente para el usuario, el contexto con el fin de interpretar la semántica proporcionada por o mostrada al usuario.

A continuación se describen las tres formas generales de sentencias INSERT estándar de SQL que pueden utilizar el diseñador o usuario experto para ingresar datos imperfectos. El módulo CAS identificará las etiquetas lingüísticas que aparezcan en la sentencia y actuará para efectuar las transformaciones lingüísticas necesarias de acuerdo a lo almacenado en el catálogo contextual (dependencias contextuales) y al contexto del usuario que interacciona.

Para ingresar una etiqueta lingüística con su semántica, la forma general de la sentencia SQL es:

```
INSERT INTO User_Table VALUES
    (... ,MC (Etiqueta, A, B, C, D, Always, ...))
```

donde,

MC es el método constructor del dominio difuso.

A, B, C, D son los valores que definen los puntos de inflexión del trapecio

Always parámetro que puede tomar el valor de 0, 1 ó 2;

De acuerdo al parámetro *Always* la definición de la etiqueta es válida:

- 0 sólo para el contexto actual,
- 1 para cualquier contexto del usuario,
- 2 para cualquier contexto de cualquier usuario.

Para ingresar datos imperfectos de una etiqueta lingüística predefinida, la forma general de la sentencia SQL es:

```
INSERT INTO User_Table VALUES (... ,MC(Etiqueta) ,...)
```

donde,

MC es el método constructor del dominio difuso.

En este caso el sistema busca en el catálogo contextual la definición válida de la etiqueta en el contexto actual. El valor recuperado como objeto trapecio es almacenado en la Base de Datos. En caso de que no se encuentre la definición de la etiqueta la BDDSC emitirá un mensaje de error.

Para Ingresar como dato imperfecto un valor, la forma general de la sentencia SQL es

```
INSERT INTO User_Table VALUES (... ,MC(Valor) ,...)
```

donde,

MC es el método constructor del dominio difuso.

Valor es un valor preciso dentro del dominio estándar (no difuso) del atributo.

En este caso el sistema ingresa directamente el valor proporcionado como un objeto trapecio en la base de datos del usuario sin necesidad de identificarlo con una etiqueta.

También prevemos que sea definida una etiqueta, sin necesidad de utilizar un método constructor que ingrese datos en una tabla de usuario, a través de un procedimiento almacenado, que se presenta en la describe a continuación.

```
DefinirEtiqueta(DomDif, Etiqueta, A, B, C, D, Always)
```

donde,

DomDif es un dominio difuso.

Etiqueta: Etiqueta lingüística.

A,B,C,D son los valores que definen los puntos de inflexión del trapecio

Always este parámetro puede tomar el valor de 0,1 ó 2 que indica: 0 si la definición de la etiqueta es válida sólo para el contexto actual, 1 si es válida para cualquier contexto del usuario ó 2 es válida para cualquier contexto de cualquier usuario.

En este caso se ingresa una etiqueta en el contexto actual de acuerdo a las dependencias definidas para el dominio difuso.

A continuación se muestran ejemplos para la utilización de estas sentencias.

Ejemplo 17. Utilización de una sentencia INSERT por *Usuario1* proporcionando una etiqueta lingüística.

Si el *Usuario1* ejecuta la siguiente sentencia INSERT:

```
INSERT INTO Exam_Tab VALUES (...,'Luis Lopez',  
D_FlexCadera('Bajo',100,115,115,125,1), 10-jun-14,...)
```

La etiqueta lingüística *Bajo* será considerada como defecto para este usuario (debido a que el parámetro *always* es 1), y es almacenada en el catálogo contextual con los valores del contexto actual de acuerdo a las dependencias contextuales del dominio difuso. Además el trapecio es almacenado para el atributo Flexión de Cadera Derecha del paciente Luis López en la tabla de Examen (*Exam_Tab*), dicho valor fue tomado el 10 de junio del 2014.

Ejemplo 18. Utilización de una sentencia INSERT por *Usuario2* proporcionando una etiqueta lingüística.

Si el *Usuario2* ejecuta la siguiente sentencia INSERT:

```
INSERT INTO Exam_Tab VALUES (...,'Ana Perez',  
D_FlexCadera('DLN',115,120,125,130,0), 12-jun-14,...)
```

La etiqueta lingüística *DLN* (Dentro de los Límites Normales) es almacenada en el catálogo contextual de con los valores del contexto actual de acuerdo a las dependencias contextuales del dominio difuso *D_FlexCadera*. Además el trapecio es almacenado para el atributo Flexión de Cadera Derecha de la paciente Ana Pérez en la tabla Examen, dicho valor fue tomado el 12 de junio del 2014.

Utilizando las sentencias estándar hasta ahora definidas para el sistema de BDDSC estos usuarios pueden definir un conjunto de etiquetas para determinado dominio difuso, tal como se muestra en los siguientes ejemplos.

Ejemplo 19. Conjunto de etiquetas de *Usuario1* para el dominio difuso Flexión de la Cadera

El *Usuario1* define un conjunto de etiquetas (*LSI*), cuya funciones de pertenencias correspondientes pueden observarse gráficamente en la Figura 27, estableciendo su semántica para la variable lingüística Flexión de la Cadera, Además, este dominio difuso es dependiente de las dimensiones contextuales *usuario* y *tarea*. En este caso el contexto para el cual se definen las etiquetas es válido para cualquier tarea.

El *Usuario2* define un conjunto de etiquetas (*LS2*), mostrado gráficamente en la Figura 28, que conforman su semántica para la variable lingüística Flexión de la Cadera, donde este dominio difuso es dependiente de las dimensiones contextuales *usuario* y *tarea*. En este caso el dominio de la dimensión contextual *tarea* es *post-operatorio*.

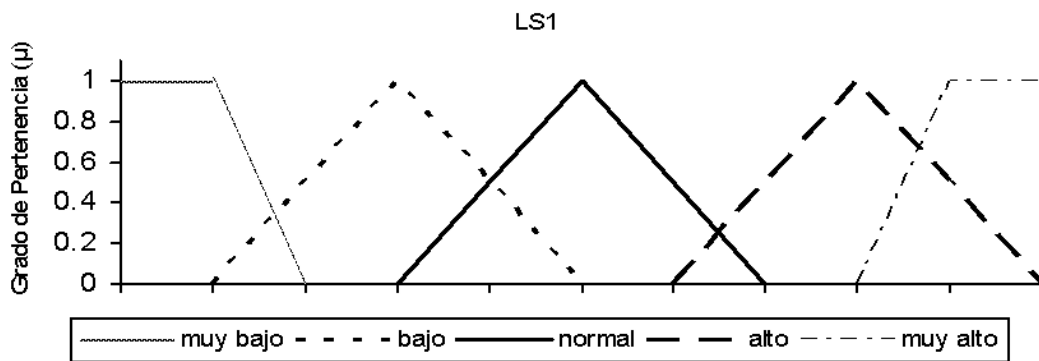


Figura 27. Conjunto de etiquetas para Flexión de la Cadera.
Contexto: *usuario1*, cualquier *tarea*

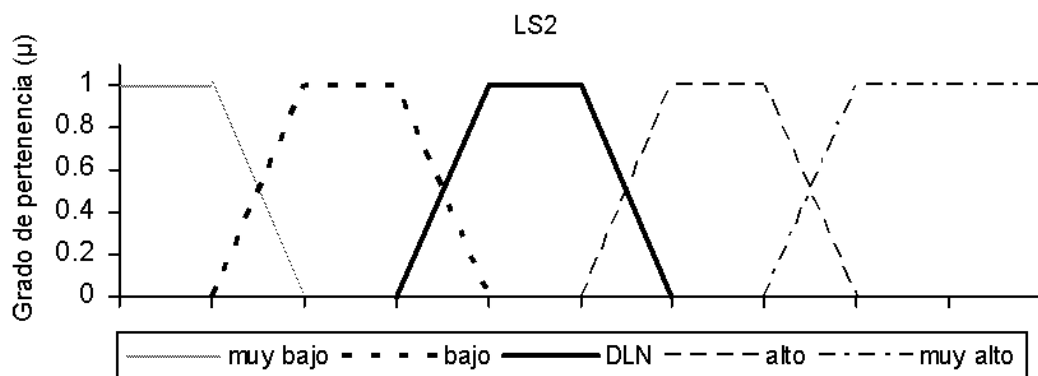


Figura 28. Conjunto de etiquetas para Flexión de la Cadera.
Contexto: *usuario2*, *postoperatorio*

Ejemplo 20. Conjunto de etiquetas de *Usuario2* para el dominio difuso Flexión de la Cadera

Es necesario recordar que estos conjuntos de etiquetas están almacenadas en el catálogo contextual, pero lo que se almacena en la base de datos son los trapecios (*datos en bruto*) de acuerdo a las sentencias INSERT ejecutadas por cada usuario. El sistema de BDDSC, a través de sus módulos CTP y con el apoyo del CASM, se encargará de hacer las transformaciones lingüísticas para presentar al usuario los datos (no en bruto) de acuerdo a la semántica definida por cada

usuario. Es por ello que a continuación presentamos ejemplos de cómo debe ser la interacción de los usuarios para efectuar consultas difusas.

5.3.2.2. Módulo de Recuperación Sensible al Contexto (CAR)

El Módulo de recuperación Sensible al Contexto (CAR) permite la recuperación de datos con semántica adaptable al usuario, mediante la utilización de métodos implementados en un SGBDOR como Oracle y los datos almacenados en el catálogo contextual para el dominio difuso atómico continuo definido en una base de datos.

Como ya se ha mencionado el módulo CAR sólo utiliza estándares de SQL:99, por lo que no se prevén problemas de costos añadidos por efectuar consultas flexibles. En todo caso se podrán hacer pruebas de rendimiento de consultas flexibles con respecto a propuestas de otros autores del área.

A continuación se presenta la descripción de los métodos utilizados por el módulo CAR.

a) Método Posiblemente Igual Difuso (FEQ)

Para nuestra BDDSC utilizaremos para el cálculo de FEQ la consideración de conjuntos difusos como distribuciones de posibilidad, donde $A(x)$ mide la posibilidad de que el dato buscado sea x en un universo de discurso X . Hay muchas formas de calcular FEQ, hemos estudiado en la sección 3.1.5 del capítulo 3 distintas formas de cálculo dependiendo del tipo de dominio difuso (ecuaciones 3.1, 3.7, 3.8, 3.13). En nuestro caso, donde utilizaremos principalmente dominios difusos atómicos continuos, la posibilidad de que el conjunto difuso A sea igual al conjunto difuso B , resulta en qué medida A y B se superponen, por lo que para el cálculo de la igualdad difusa $FEQ(A,B)$ utilizamos la ecuación 3.7 con el mínimo como t-norma.

$$FEQ(A, B) = \sup_{x \in X} [\min(A(x), B(x))] \quad (5.1)$$

La aplicación de esta definición para dos conjuntos difusos atómicos continuos, representados por funciones de pertenencia trapezoidales, se muestra gráficamente en la figura 29. Dependiendo de la t-norma utilizada, el FEQ puede ser más o menos restrictivo. En el caso en particular tratado en esta investigación la forma de calcular este operador puede ser dependiente del atributo, por ejemplo se puede aplicar una definición más restrictiva para un atributo como el tipo de marcha patológica y menos restrictiva para comparar el peso de dos personas,

aunque en nuestra implementación utilizamos sólo la ecuación 5.1 definida anteriormente.

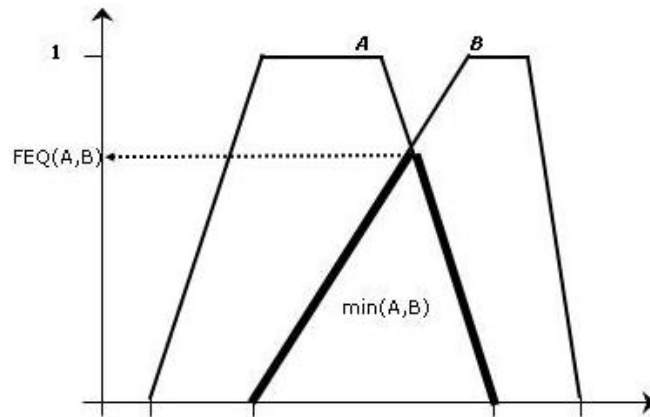


Figura 29. Ilustración del cálculo de *Fuzzy Equal (FEQ)*

b) Método Mostrar Etiquetas (LShow)

Por la importancia que reviste en nuestra propuesta a continuación definimos formalmente la *expresión lingüística*, que nos servirá luego para implementar el método de mostrar etiquetas (LShow).

Definición 12. Expresión Lingüística

Sean,

$LS = \{Etiqueta_1^{LS}, \dots, Etiqueta_n^{LS}\}$ un conjunto de etiquetas lingüísticas definido para el dominio difuso D , y

fs Conjunto difuso definido para el dominio difuso D .

Entonces, la Expresión Lingüística (LE) de fs en términos del conjunto de etiquetas LS , $LE_{LS}(fs)$ se calcula de la siguiente forma:

$$LE_{LS}(fs) = \mu_1 Etiqueta_1^{LS} + \dots + \mu_n Etiqueta_n^{LS}$$

donde,

$$\mu_i = \Psi(fs, Etiqueta_i^{LS}), \forall i = 1, \dots, n$$

y Ψ es un operador de semejanza como los definidos en [90] [91].

Mediante esta operación combinamos las etiquetas lingüísticas representadas por conjuntos difusos atómicos continuos, de representaciones semánticas diferentes de una variable lingüística para producir conjuntos difusos de segundo nivel.

Por ejemplo, si el *usuario1* tiene tres etiquetas lingüísticas para el peso de una persona (*delgado*, *promedio* y *gordo*), definidas en un contexto dado, las cuales son descritas por funciones de pertenencia trapezoidales, éstas son comparadas con la función de pertenencia tipo trapezoidal almacenada en la base de datos en el atributo peso de cada persona.

Esta es la regla de inferencia utilizada por el módulo de Recuperación Sensibles al Contexto (CAR), utilizando para ello el método de comparación posiblemente igual difuso (FEQ) definido para los objetos tipo trapecio, de la forma que se explica a continuación.

Si el usuario *I* define *n* conjuntos difusos atómicos continuos (L_1, L_2, \dots, L_n) para una variable lingüística *X* en un contexto dado; además, existe una tupla en la Base de Datos donde el atributo *X* es descrito por la función de pertenencia tipo trapezoidal *B*, entonces el conjunto difuso *C*, es definido por la Ec. 5.2.

$$C = \mu_1 L_1 + \mu_2 L_2 + \dots + \mu_n L_n \quad (5.2)$$

donde,

$$\mu_i = FEQ(L_i, B) \quad \text{para } i=1,2,\dots,n$$

Ejemplo 21. Utilización del método LSHOW por parte del *Usuario1*.

Si el *Usuario1* ejecuta la siguiente consulta SQL, sin importar el valor de la dimensión contextual *tarea*.

```
SELECT Name, ET.FlexCadDer.LShow() FlexCaderaDerecha
FROM Exam_Tab ET
WHERE Name='Ana Perez'
```

Obtiene la respuesta:

NAME	FlexCaderaDerecha
-----	-----
Ana Perez	0.7 Bajo; 1 Normal;

El módulo CTP de Recuperación Sensible al Contexto (CAR) aplicó la Transformación Lingüística adecuada para presentar al *Usuario1* la información almacenada en la Base de datos para “Ana Perez” por el *Usuario2* (en forma de trapecio) con la semántica establecida por el *Usuario1* (LS1) para el contexto.

En la figura 30 se muestra gráficamente este resultado, donde el dato en bruto almacenado para Ana Perez (representado por la etiqueta DLN de *Usuario2*) es transformado a la semántica establecida en el contexto por el *Usuario1* (LS1).

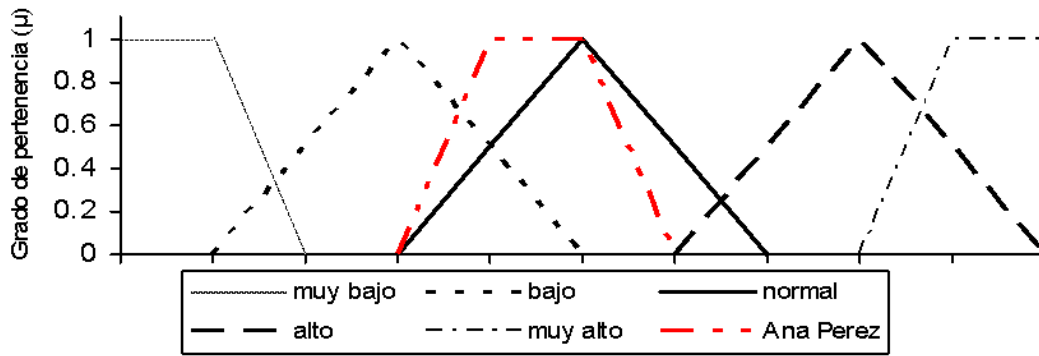


Figura 30. Transformación Lingüística para el *Usuario1*

Ejemplo 22. Utilización del método LSHOW por parte del *Usuario2*.

Si el *Usuario2* ejecuta la siguiente consulta SQL, en un contexto donde la dimensión contextual *tarea* tiene el valor *post-operatorio*.

```
SELECT Name, ET.FlexCadDer.LShow() FlexCaderaDerecha
FROM Exam_Tab ET
```

Obtiene la respuesta:

NAME	FlexCaderaDerecha
-----	-----
Luis Lopez	0.3 Muy Bajo; 1 Bajo; 0.7 DLN;
Ana Perez	0.5 Bajo; 1 DLN; 0.5 Alto;

Ahora se presenta la información almacenada en la Base de Datos con la semántica establecida por el *Usuario2* (LS2) de acuerdo al contexto establecido. Es por ello que denominamos a esta transformación lingüística “Semántica Adaptable”, dado que a pesar que los datos en bruto almacenado son los mismos, las respuestas se ajustan a la semántica definida por cada usuario en un contexto específico.

En la figura 31 se muestra gráficamente este resultado, donde los datos en bruto almacenado en la Tabla Examen son transformados a la semántica establecida en el contexto por el *Usuario2* (LS2).

De esta forma logramos implementar lo que denominamos en nuestro modelo un enfoque “flexible de sensibilización al contexto”, en contraposición del tradicional paradigma de “coincidencia exacta”.

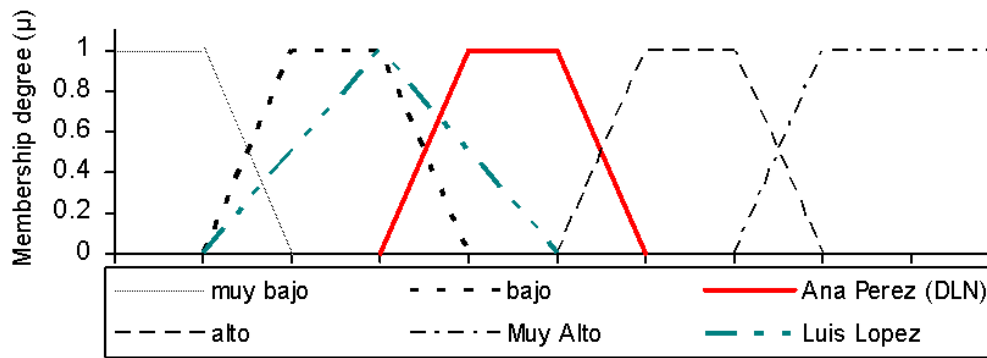


Figura 31. Transformación Lingüística para el *Usuario2*

5.4. Conclusiones del Capítulo

Mediante técnicas de representación y manipulación de objetos imperfectos, utilizando el modelo teórico propuesto, presentamos una particularización que permite construir una base de datos cuya semántica de términos descriptivos se adapta al usuario, evitándole tener que definir esa semántica cada vez que difiere de la que originalmente es definida por el diseñador del sistema.

En el próximo capítulo se presenta la prueba concepto en el área médica, donde el usuario podrá interactuar con el SBDDSC utilizando sentencias del lenguaje de consultas estándar SQL de un SGBDOR tal como se describieron en este capítulo.

Para el mismo dominio del área médica se diseñó una aplicación web con interfaz móvil, permitiendo que usuarios inexpertos que no conocen SQL, interactúen en forma amigable y transparente con el SBDDSC, donde en capa física de la Base de Datos Difusa se implementan las primitivas necesarias para utilización del modelo propuesto.

Capítulo 6

Aplicación Real de un SBDDSC

Una vez establecido el modelo de Sistema de Bases de Datos Difusas Sensible al Contexto (SBDDSC) y las bases formales para su desarrollo, en este capítulo mostraremos una prueba de concepto mediante una implementación en SQL utilizando las capacidades de un SGBDOR tal como Oracle y una aplicación web que sirva de mediador entre usuarios inexpertos y el SBDDSC.

6.1. Descripción de la Aplicación Real en SQL

La prueba de concepto se efectuó para el Hospital Ortopédico Infantil de Venezuela. En este Hospital se instaló en 1998 un Laboratorio de Análisis de Marcha como una unidad clínica de apoyo al diagnóstico de pacientes referidos por otros centros hospitalarios o médicos particulares. Nos enfocamos en el examen físico articular que se realiza en el laboratorio en marcha del Hospital. En el Apéndice A se muestra una descripción de las interfaces del sistema actual que se utilizan y los datos que se necesitan recolectar.

Hay que destacar que la base de datos del examen Físico Articular es única, la asociación entre etiquetas y contexto se realiza de forma distribuida durante el uso de la base de datos. El diseñador sólo define valores por defecto y las dependencias contextuales al crear las tablas. A continuación se describe el dominio de la aplicación.

6.1.1. Dominio de la Aplicación

La necesidad de un sistema sensible al contexto la hemos encontrado en nuestro trabajo en relación con el manejo de datos médicos en un ambiente multidisciplinar en el que intervienen distintos tipos de usuarios que interactúan con los mismos datos. El área médica es un ejemplo donde las decisiones están basadas en datos imperfectos, además de jugar un rol vital el contexto. En la práctica, cualquier decisión médica incluye un cierto grado de subjetividad basada en hechos determinados por diversos medios; es así como el médico debe efectuar

diagnósticos y aplicar tratamientos tomando en cuenta diferentes variables tales como: edad, estado emocional, estado físico, antecedentes del paciente; además, las decisiones tomadas por un médico deben ser oportunas y contando con información fiable, ya que en muchos casos, está en juego el bienestar de los pacientes.

En particular, nos enfocaremos en el problema del estudio de la marcha humana, donde se cuantifica el comportamiento articular, músculo-esquelético y se reflejan las secuelas del compromiso neuromuscular de los pacientes con marchas patológicas (Parálisis Cerebral, Espina Bífida, Mielodisplasia, malformaciones congénitas, entre otras).

En la figura 32 se presenta un esquema general del ejemplo, donde se muestran los diferentes usuarios que pueden interactuar con el sistema y los múltiples contextos generados en base a la semántica particular de los términos utilizados en el sistema.

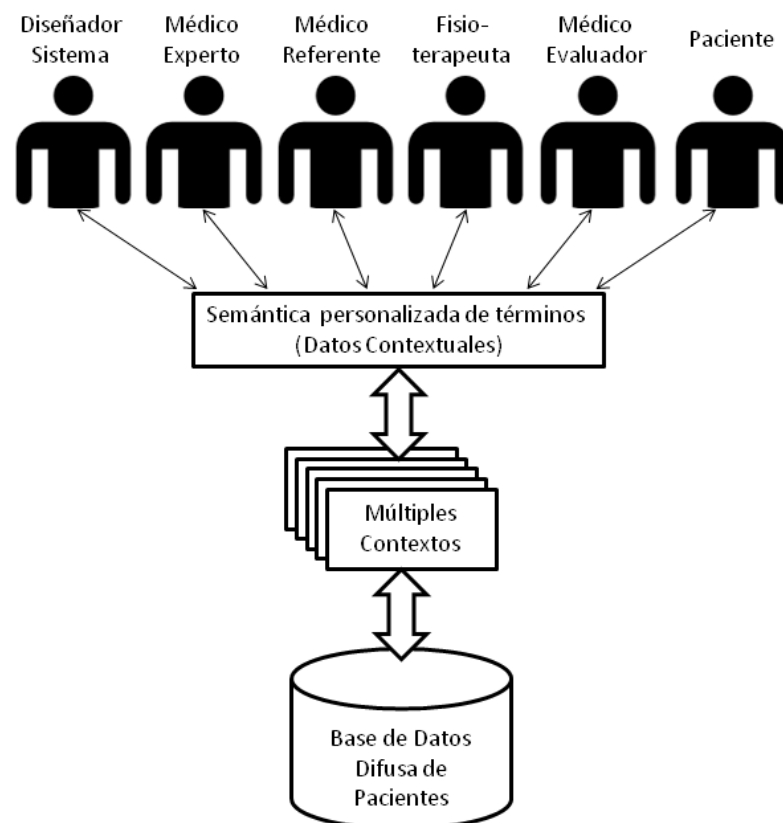


Figura 32. Esquema General de la Aplicación Real

- El diseñador del sistema establece los metadatos contextuales, es decir, el patrón por el cual se va a describir y gestionar los datos contextuales.
- Un médico experto puede definir semántica de términos lingüísticos, por ejemplo, el peso normal de una persona.

- El médico referente realiza un diagnóstico del paciente analizando síntomas y posibles enfermedades, el cual decide enviar al paciente a un Laboratorio en Marcha para un estudio completo en el caso de disfunciones neuromotoras.
- El Fisioterapeuta realiza un examen físico articular, evaluando las características biomecánicas en las articulaciones de los miembros inferiores del paciente (cadera, rodilla, tobillo y pie) y el movimiento característico asociado a cada una de ellas.
- El Médico evaluador utiliza estos datos junto con los datos personales y otros estudios para efectuar el diagnóstico.

Cada usuario que interviene puede establecer su semántica particular a través de etiquetas lingüísticas de datos imperfectos que va a asociar a diversos contextos, tales como: rangos articulares, selectividad (capacidad de realizar un movimiento), fuerza muscular, tono muscular o reflejos osteotendinosos. Esto genera múltiples contextos, donde cada uno de ellos está ligado a un usuario particular, representando la personalización de la semántica.

Es de hacer notar que existen diversos factores que pueden influir en la objetividad de la medición articular, tales como: variabilidad del individuo que efectúa la medición, el instrumento de medición utilizado, la notación, el conocimiento anatómico y la estimación de los valores normales de las diferentes articulaciones. La base de datos de pacientes puede contener datos difusos, ya sea en su representación o a nivel de consulta; tales como los anteriormente mencionados u otros relativos al paciente (peso, talla o edad).

En las próximas secciones se demuestra la factibilidad de llevar a cabo el modelo propuesto en un caso médico concreto: el examen físico articular en un laboratorio en marcha. Se tendrán diversos usuarios que puedan interactuar directamente con el sistema a través del estándar SQL:99, prescindiendo de la necesidad de aplicaciones de gestión sensibles al contexto; no se dispondrá de sensores ambientales, ni agentes que puedan inferir datos contextuales.

6.1.2. El Examen Físico Articular

Dentro de los análisis realizados en este hospital se efectúa un examen físico articular que evalúa las características biomecánicas observadas en las articulaciones de los miembros inferiores de los pacientes: la fuerza y el tono muscular (tomados por apreciación del evaluador), los rangos de movimientos articulares y las probables deformidades asociadas. En este examen se generan datos imperfectos debido a la multiplicidad de factores que influyen en la objetividad de la medición articular: Variabilidad que depende de la persona que efectúa la medición, instrumento de medición (Goniómetro, centímetro,

instrumento electrónico), notación utilizada por diferentes usuarios (lenguaje de escritura), conocimiento anatómico y la estimación de los valores normales de las diferentes articulaciones (por ejemplo, flexión de cadera de 120 a 130 grados).

Los datos del examen físico articular representan 143 valores distintos para cada paciente evaluado. Aunque las evaluaciones corresponden en gran medida a valores precisos medidos por un especialista de fisioterapia, en algunos casos estos valores provienen de la observación del movimiento; además, se utilizan etiquetas lingüísticas tales como “Dentro de los Límites Normales (DLN)”, donde el especialista puede definir estas etiquetas como rangos para cada articulación según sus preferencias. El proceso seguido por el Hospital Ortopédico Infantil se muestra en el diagrama de caso de uso de la Figura 33 y se describe a continuación.

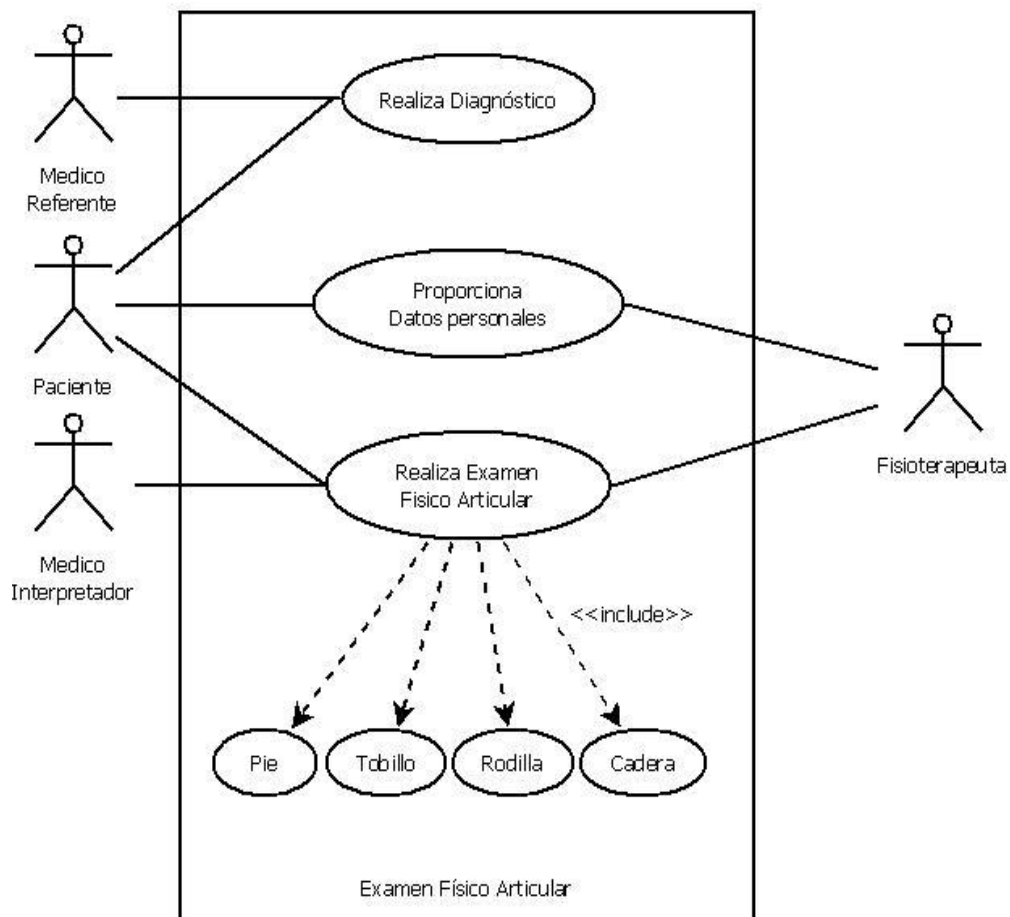


Figura 33. Caso de Uso Examen Físico Articular, Hospital Ortopédico Infantil

El contexto (por ejemplo usuario y tarea) puede determinar la variabilidad de valores normales. Además debido a la diversidad de usuarios y roles (diferentes médicos especialistas, fisioterapeuta, pacientes, familiares) el

significado de etiquetas lingüísticas en variables como el peso de una persona puede variar.

El médico referente realiza un diagnóstico del paciente analizando síntomas y posibles enfermedades, decide enviar al paciente al Laboratorio en Marcha para un estudio completo en el caso de disfunciones neuromotoras (Parálisis Cerebral, Espina Bífida, Mielodisplasia, malformaciones congénitas, entre otras); el fisioterapeuta evalúa las características biomecánicas en las articulaciones (cadera, rodilla, tobillo y pie) de los miembros inferiores del paciente y el movimiento característico asociado a cada una de ellas.

Por ejemplo, la cadera es una articulación denominada enartrosis debido a que realiza todos los movimientos articulares, tales como: flexión, extensión, abducción, aducción, rotación interna, rotación externa. Se evalúa en particular el rango articular, selectividad/fuerza muscular y tono muscular en ambos miembros. El Médico Interpretador utiliza estos datos junto con los datos personales y otros estudios para efectuar el diagnóstico.

En la Figura 34 se presenta el diagrama de caso de uso del sistema propuesto de Bases de Datos Difusas Sensibles al Contexto para el examen Físico Articular del Hospital Ortopédico Infantil. A continuación una breve descripción de las funcionalidades del sistema.

El diseñador de la base de datos efectúa la creación de clases para definir dominios difusos, crea las tablas que utilizan dominios difusos y sinónimos que permitan utilizar las tablas por todos los usuarios, compila funciones y procedimientos de PL/SQL; todo esto utilizando scripts y plantillas ya establecidos en la particularización explicada en la sección anterior. Además, ingresa datos de los dominios difusos, usuario por defecto y etiquetas lingüísticas definidas por un médico experto.

El Fisioterapeuta registra los datos en la base de datos difusa utilizando los métodos y constructores de las clases, o en su defecto aplicaciones que sirvan de interfaz, de acuerdo a los resultados del examen Físico-Articular realizado al paciente.

El Médico Interpretador x ($x=1, \dots, n$) realiza consultas a la Base de Datos Difusa basadas en el contexto, utilizando la definición de etiquetas lingüísticas de acuerdo a un contexto. También es posible que otros usuarios como el mismo paciente o familiares puedan realizar consultas difusas con semántica adaptable sensibles al contexto.

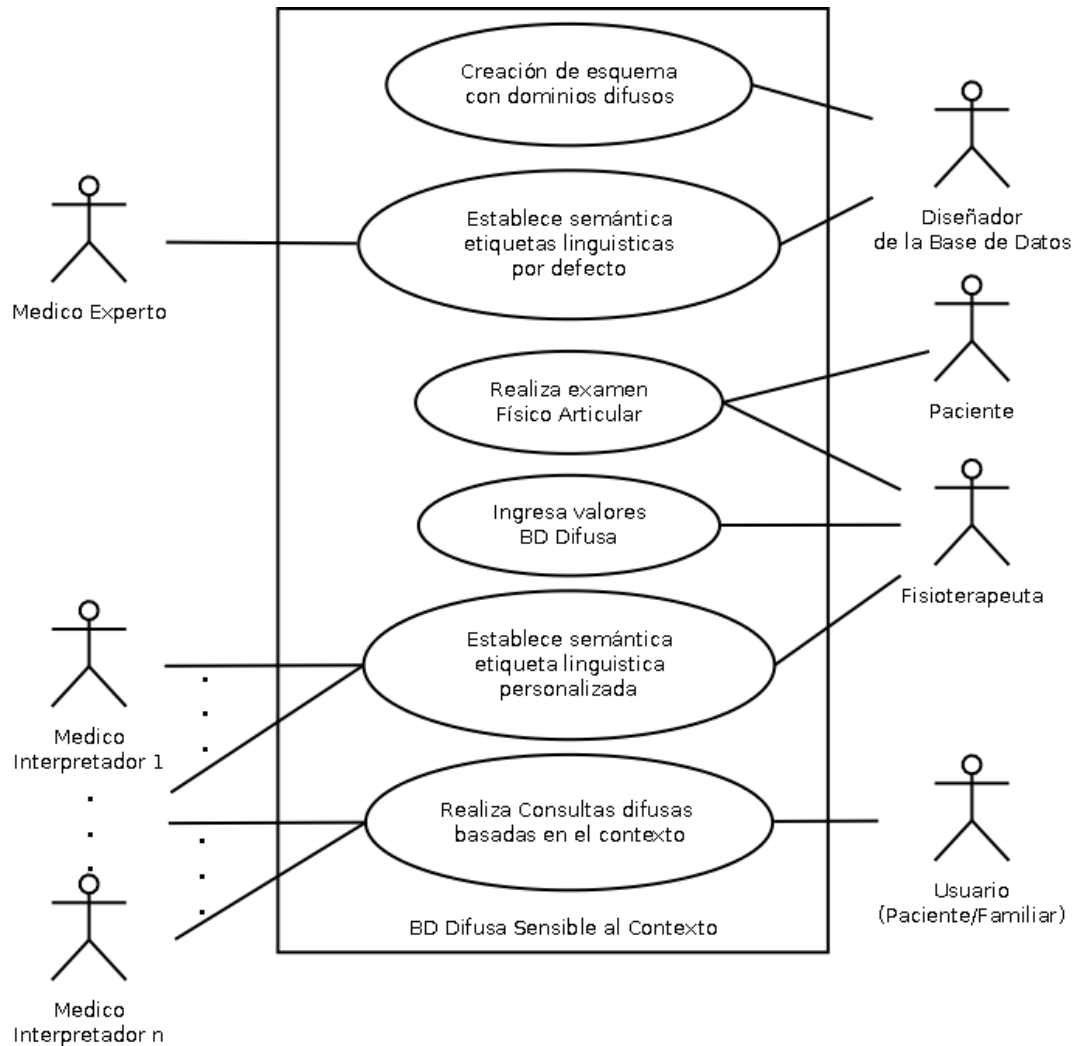


Figura 34. Aplicación Real de BD Difusa Sensible al Contexto

En nuestro caso los usuarios que interactúan en el sistema son: diseñador de la base de datos, médicos especialistas, fisioterapeutas, pacientes y/o familiares; así que se debe facilitar una forma amigable para que estas personas explícitamente ingresen a través del lenguaje de consulta estándar SQL99 del SGBD Oracle operaciones relativas al contexto, representando la semántica de los términos utilizados por cada usuario (posiblemente diferente). Esto no limita el hecho de que en futuras investigaciones, se logre la interoperación con sensores ambientales o agentes de software para obtener entradas contextuales implícitas, logrando una aplicación de base de datos difusa sensible al contexto en un sentido más amplio.

Es así como se requiere almacenar datos personales de los pacientes y valores obtenidos de exámenes tal como el físico articular (el cual se explica más detalladamente en la próxima sección).

En la figura 35, se presenta un diagrama de clases donde se tiene una versión reducida de los datos almacenados para dicho examen y la extensión del dominio difuso atómico continuo para atributos como el peso (D_Peso) y la flexión de la cadera (D_FlexCadera).

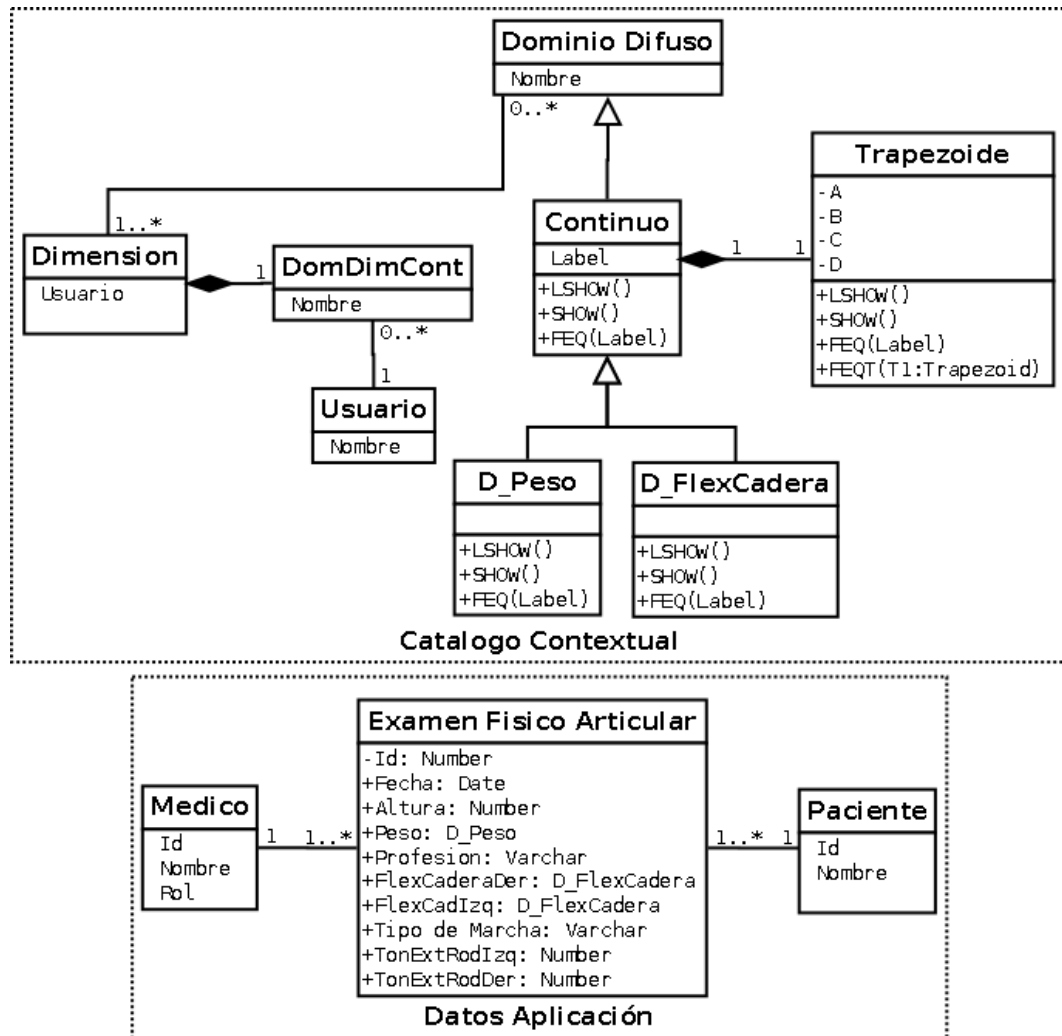


Figura 35. Diagrama de Clases para la implementación de la Aplicación Real

Es de hacer notar que la Flexión de la Cadera se propone como un atributo cuyo valor puede ser difuso, ya que el especialista puede colocar la etiqueta lingüística DLN (Dentro de Límites Normales) la cual está definida por un trapecio, que puede ser distinto para cada usuario. Además, el peso se diseñó también como dominio difuso, ya que a pesar de ser obtenido en forma precisa por un fisioterapeuta, a nivel de consultas se pueden utilizar etiquetas como “obeso”, “delgado” o “normal”; definidas por trapecios los cuales pueden ser diferentes para cada usuario del sistema.

En este caso, el sistema identifica al usuario al conectarse el mismo a la base de datos para efectuar inserciones o consultas. A pesar que en nuestro modelo el contexto puede tener otras dimensiones contextuales cómo por el ejemplo la tarea que el usuario está realizando (examen preoperatorio, postoperatorio, general) y dominios para esta dimensión, en el ejemplo sólo utilizamos como dimensión contextual al usuario.

La semántica de la etiqueta “DLN” para un rango articular es sensible al contexto, así como cualquier otra etiqueta lingüística tal como “obeso” para definir el peso de una persona. Si un usuario utiliza una etiqueta que no es definida por él, se recurre a la definida por el usuario por defecto para el contexto en cuestión. Esto constituye la “inteligencia” del sistema al sensibilizar la semántica de las etiquetas lingüísticas de acuerdo al contexto del usuario conectado.

Finalmente, en la base de datos se almacenan datos adaptados en base al contexto por la semántica definida por cada usuario (o la del usuario por defecto), para atributos definidos por tipos de datos con dominios difusos. El sistema está en capacidad de registrar en el catálogo contextual de la meta base de conocimiento: por cada dominio difuso y contexto (usuario) la definición de las etiquetas lingüísticas.

En la próxima sección se muestra la implementación del ejemplo detallado utilizando el estándar SQL99, con datos de muestra para una mejor comprensión de lo explicado hasta este momento.

6.1.3. Implementación de la Aplicación real en SQL

A continuación presentaremos la implementación del ejemplo utilizando sentencias estándar de SQL:99 utilizando el Sistema Gestor de Bases de Datos Oracle *Express Edition* Versión 11.2 y estructuras estándares del lenguaje PL/SQL.

6.1.3.1. Catálogo Contextual

En la figura 36 se puede ver una muestra de valores almacenados en el catálogo contextual. Cada usuario puede personalizar la semántica de las etiquetas lingüísticas utilizadas para cada tipo de dato definido como un dominio difuso atómico continuo (en este caso el peso y la flexión de la cadera), representado por un trapecio definido con cuatro parámetros (A , B , C , D), donde A y D representan el soporte, B y C el núcleo de la función de pertenencia.

Etiqueta	DomCtx	DomDifuso	Trapezio
DLN	USERDEF	FlexCad	TRAPECIO (110, 120, 130, 140)
Alto	USERDEF	FlexCad	TRAPECIO (130, 150, 360, 360)
Bajo	USERDEF	FlexCad	TRAPECIO (0, 0, 100, 120)
Normal	USERDEF	Peso	TRAPECIO (80, 90, 100, 110)
Obeso	USERDEF	Peso	TRAPECIO (100, 110, 300, 300)
Delgado	USERDEF	Peso	TRAPECIO (0, 0, 80, 90)
DLN	USER1	FlexCad	TRAPECIO (120, 130, 140, 150)
DLN	USER2	FlexCad	TRAPECIO (110, 125, 145, 160)
Bajo	USER1	FlexCad	TRAPECIO (0, 0, 120, 130)
Alto	USER1	FlexCad	TRAPECIO (140, 160, 360, 360)
Bajo	USER2	FlexCad	TRAPECIO (0, 0, 110, 125)
Alto	USER2	FlexCad	TRAPECIO (145, 160, 360, 360)

Figura 36. Muestra de valores almacenados en el Catalogo Contextual

Cuando un usuario ingrese datos en la tabla del Examen Físico Articular, el Modulo CTP de Almacenamiento Sensible al Contexto (CAS) actúa para utilizar la semántica particular del usuario (o en todo caso la del usuario por defecto). Cuando otro usuario consulte estos datos, el módulo CTP de Recuperación Sensible al Contexto (CAR) se encargará de presentar la información traducida a la semántica definida por el usuario (o en la del usuario por defecto) que hace la consulta; para ello, este módulo utiliza la lógica adecuada almacenada en la meta base de conocimiento de las *reglas de inferencia* (transparentes para el usuario final), en este caso representadas por las combinaciones convexas de las diferentes etiquetas definidas por el usuario para una variable lingüística como el peso o la flexión de la cadera.

6.1.3.2. Definición y manipulación de datos a través de SQL99

Para demostrar las bondades del sistema propuesto a continuación se presenta el ejemplo implementado en una tabla reducida donde se realiza la creación, la inserción y las consultas.

a) Creación de tabla Examen Físico Articular.

En este caso sólo se registra un identificador de persona, nombre, talla, peso, profesión, flexión cadera derecha, flexión cadera izquierda, tipo de marcha, tono extensor rodilla izquierda y tono extensor rodilla derecha en la tabla denominada EFA_tab de acuerdo a lo diseñado en la sección anterior.

```

CREATE TABLE EFA_tab (
  Id_Person      NUMBER  Constraint PK_Per PRIMARY KEY,
  Nombre         VARCHAR2(20),
  Talla          NUMBER,
  Peso           D_Peso,
  Profesion      VARCHAR2(20),
  FlexCadDer     D_FlexCad,
  FlexCadIzq     D_FlexCad,
  TipoMarcha     VARCHAR2(20),
  TonExtRodIzq  NUMBER,
  TonExtRodDer   NUMBER
);

```

b) Ingreso de datos.

1. Si el usuario por defecto ejecuta los siguientes ingresos en la base de datos:

```

INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Phillip
Rousell', 167, D_Peso(85), 'Investigador', D_FlexCad('DLN'),
D_FlexCad(115), 'Normal', 1, 4);

```

```

INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Pier Casteran',
160, D_Peso(102), 'Profesor', D_FlexCad(110),
D_FlexCad('DLN'), 'Patologica', 2, 3);

```

El Modulo CTP de Almacenamiento Sensible al Contexto (CAS) actúa para además de ingresar datos de la persona en la tabla del Examen Físico Articular, tomar los valores definidos por el usuario por defecto en el catálogo contextual para ingresar los trapecios correspondientes en “DLN”; en los casos como D_Peso(85) y D_FlexCad(115) se insertan trapecios que corresponden a valores precisos.

2. Si *User1* ejecuta los siguientes ingresos en la base de datos:

```

INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Jose Castillo',
175, D_Peso(100), 'Deportista',
D_FlexCad('DLN',120,130,140,150), D_FlexCad(155), 'Normal',
4, 4);

```

```

INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Cesar Rondon',
170, D_Peso(90), 'Locutor', D_FlexCad('Bajo',0,0,120,130),
D_FlexCad('Alto', 140,160,360,360), 'Parkinson', 1, 1);

```

```

INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Maria Rivas',
177, D_Peso('Normal'), 'Cantante', D_FlexCad('DLN'),
D_FlexCad('Bajo'), 'Normal', 4, 4);

```

El módulo CAS ingresa los datos de la persona, además de en las definiciones para las etiquetas DLN, Alto y Bajo (para el dominio difuso D_FlexCad) y Normal (para el dominio difuso D_Peso) de acuerdo al contexto, en este caso el usuario.

3. Si un usuario, por ejemplo *User2*, ejecuta los siguientes ingresos en la base de datos:

```
INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Jose Bardina',
180, D_Peso(99), 'Actor', D_FlexCad('DLN',110,125,145,160),
D_FlexCad(155), 'Normal', 4, 4);
```

```
INSERT INTO EFA_tab VALUES (Id_Per.nextval, 'Luis Lopez',
170, D_Peso(120), 'Electricista',
D_FlexCad('Bajo',0,0,110,125), D_FlexCad('Alto',
145,160,360,360), 'Ataxica', 1, 1);
```

Se ingresan datos de la persona, además de en las definiciones para las etiquetas (DLN, Alto y Bajo) en el catálogo contextual para el usuario.

c) Consultas

1. Si se desea ver la lista de personas en la tabla EFA_tab con la semántica de un usuario particular (por ejemplo *User1*), el Módulo CTP de Recuperación Sensible al Contexto (CAR), utiliza las reglas de inferencia programadas a través de procedimientos almacenados y métodos de los dominios difusos para obtener un conjunto difuso de acuerdo a la semántica particular del usuario.

```
SELECT Nombre, E.FlexCadDer.LShow()
Flexion_Cadera_Derecha, E.FlexCadIzq.LShow()
Flexion_Cadera_Izquierda
FROM EFA_tab E;
```

NOMBRE	FLEXION_CADERA_DER	FLEXION_CADERA_IZQ
Phillip Rousell	1/DLN; 1/Bajo;	
Pier Casteran		1/DLN; 1/Bajo;
Jose Castillo	1/DLN; 0,5/Bajo; 0,33/Alto;	0,75/Alto;
Maria Rivas	1/DLN; 0,5/Bajo; 0,33/Alto;	1/Bajo;
Jose Bardina	1/DLN; 0,8/Bajo; 0,57/Alto;	0,75/Alto;
Cesar Rondon	0,5/DLN; 1/Bajo;	0,33/DLN; 1/Alto;
Luis Lopez	0,2/DLN; 1/Bajo;	0,2/DLN; 1/Alto;

2. La misma consulta efectuada por otro usuario (*User2*), obtiene el siguiente resultado:

NOMBRE	FLEXION_CADERA_DER	FLEXION_CADERA_IZQ
Phillip Rousell	1/DLN; 0,6/Bajo;	0,33/DLN; 0,67/Bajo;
Pier Casteran		1/DLN; 0,6/Bajo;
Jose Castillo	1/DLN; 0,2/Bajo; 0,2/Alto;	0,33/DLN; 0,67/Alto;
Maria Rivas	1/DLN; 0,2/Bajo; 0,2/Alto;	0,29/DLN; 1/Bajo;
Jose Bardina	1/DLN; ,5/Bajo; 0,5/Alto;	0,33/DLN; 0,67/Alto;
Cesar Rondon	0,8/DLN; 1/Bajo;	0,57/DLN; 1/Alto;
Luis Lopez	0,5/DLN; 1/Bajo;	0,5/DLN; 1/Alto;

Es importante resaltar que a pesar de mantener los mismos datos en bruto el sistema está ofreciendo la información de acuerdo al conjunto de términos (o semántica) dada por cada usuario a las etiquetas lingüísticas para el dominio difuso de la flexión de cadera.

3. Consulta para ver la etiqueta “DLN” (Dentro de los Límites Normales) definida por el usuario con respecto a los datos en la tabla del EFA (User1). Aquí se utiliza el método FEQ definido como regla de inferencia para el dominio difuso Flexión de Cadera.

```
SELECT Nombre, E.FlexCadDer.FEQ('DLN') FEQ
FROM EFA_tab E
WHERE E.FlexCadDer.FEQ('DLN')>0;
```

NOMBRE	FEQ
Phillip Rousell	1
Jose Castillo	1
Maria Rivas	1
Jose Bardina	1
Cesar Rondon	0,5
Luis Lopez	0,2

Como *User2* y la misma consulta, el módulo CAR actúa de forma inteligente para obtener el siguiente resultado:

NOMBRE	FEQ
Phillip Rousell	1
Jose Castillo	1
Maria Rivas	1
Jose Bardina	1
Cesar Rondon	0,8
Luis Lopez	0,5

4. Luego para hacer consultas para mostrar flexión de cadera de forma más amigable, aquí se obtienen los datos en bruto almacenados en la Base de Datos del Examen Físico Articular.

```
SELECT Nombre, E.FlexCadDer.Show() Flexion_Cadera_Derecha,
        E.FlexCadIzq.Show() Flexion_Cadera_Izquierda
FROM EFA_tab E;
```

NOMBRE	FLEXION_CADERA_DER	FLEXION_CADERA_IZQ
Phillip Rousell	(110, 120, 130, 140)	(115, 115, 115, 115)
Pier Casteran	(110, 110, 110, 110)	(110, 120, 130, 140)
Jose Castillo	(120, 130, 140, 150)	(155, 155, 155, 155)
Maria Rivas	(120, 130, 140, 150)	(0, 0, 100, 120)
Jose Bardina	(110, 125, 145, 160)	(155, 155, 155, 155)
Cesar Rondon	(0, 0, 120, 130)	(140, 160, 360, 360)
Luis Lopez	(0, 0, 110, 125)	(145, 160, 360, 360)

5. Para obtener las personas que tienen Flexión de Cadera Derecha “DLN”. El modulo CAR actúa de nuevo utilizando la regla de inferencia FEQ como atributo y como predicado dentro de la cláusula WHERE.

```
SELECT Nombre, E.FlexCadDer.FEQ('DLN') FEQ
FROM EFA_tab E
WHERE E.FlexCadDer.FEQ('DLN')>0;
```

NOMBRE	FEQ
Phillip Rousell	1
Jose Castillo	1
Maria Rivas	1
Jose Bardina	1
Cesar Rondon	0,8
Luis Lopez	0,5

6. Para obtener las personas obesas que tengan Flexión de Cadera Derecha DLN, el modulo CAR utiliza en este caso el método FEQ para los dominios difusos D_Flexión de Cadera y Peso.

```
SELECT Nombre, E.FlexCadDer.FEQ('DLN') FEQ
FROM EFA_tab E
WHERE E.Peso.FEQ('Obeso')>0;
```

NOMBRE	FEQ
Pier Casteran	0
Maria Rivas	1
Luis Lopez	,5

7. El mismo caso anterior pero ahora el módulo CAR identifica las personas de peso normal que tengan Flexión de Cadera Derecha DLN.

```
SELECT Nombre, E.FlexCadDer.FEQ('DLN') FEQ
FROM EFA_tab E
WHERE E.Peso.FEQ('Normal')>0;
```

NOMBRE	FEQ
Phillip Rousell	1
Pier Casteran	0
Jose Castillo	1
Maria Rivas	1
Jose Bardina	1
Cesar Rondon	,8

Como se puede notar claramente, de acuerdo a la semántica dada por cada usuario, los módulos CTP de sensibilidad al contexto actúan de forma inteligente utilizando el catalogo contextual y los métodos definidos para cambiar la

interpretación semántica aunque se está utilizando el mismo conjunto de datos almacenados en la base de datos. También es importante desatacar que mediante métodos como *LSHOW* se muestra al usuario los datos según su semántica (basado en sus preferencias).

6.2. Diseño de la Aplicación Web

Una aplicación web sensible al contexto puede gestionar datos contextuales (los almacena, modifica y utiliza), los cuales pueden ser implícitos y explícitos. Los explícitos están asociados a un perfil de un usuario determinado, permitiendo tomarlos como insumos para que el cambio de su valor pueda afectar la salida de una consulta realizada ya sea en su semántica, en su resultado o en su presentación. Con la utilización de bases de datos difusas en el *back-end* para la representación de datos imperfectos, permitimos agregarle un mayor valor semántico al sistema, además de poder efectuar consultas difusas; de esta forma se puede obtener una aproximación más cercana a los usuarios, que estos se sientan identificados con la manera en la que se manejan los datos

Las herramientas tecnológicas que se utilizaron para el desarrollo de la aplicación web sensible al contexto usando bases de datos difusas y su interfaz móvil se describen a continuación.

6.2.1. Plataforma de Desarrollo

La aplicación Web fue desarrollada utilizando una plataforma XAMPP (Linux, Apache, MySQL, PHP, Perl), la cual es una distribución gratuita de fácil instalación y configuración, que utiliza el servidor web Apache, donde funciona el sistema implementado en PHP con complementos OCI para la integración de Oracle.

Para la elaboración de la Interfaz Web se usó el lenguaje de marcado HTML5. En cuanto al diseño de la interfaz móvil se aprovecharon distintos componentes de *Bootstrap*, un *framework* de diseño de aplicaciones web para dispositivos móviles, el cual contiene componentes prediseñados que se adaptan al tamaño del dispositivo (*responsive*). En particular, se usaron botones sensibles al tamaño de la pantalla, alertas predeterminadas para mantener la coherencia, iconos para presentar las opciones disponibles al usuario, conjuntos de formularios para simplificar la obtención de información, entre otros. *Bootstrap* está especificado en HTML, CSS, LESS y Javascript.

6.2.2. Lenguajes de Programación

Las funcionalidades fueron desarrolladas usando los lenguajes de programación: PHP y JQuery. PHP fue utilizado para las conexiones con la base de datos desde el sistema y para el manejo de datos y consultas. JQuery es una librería de JavaScript que se utilizó para la realización de calendarios, efectos visuales sencillos y para la elaboración de gráficas de los datos.

6.2.3. Base de Datos

El gestor de bases de datos utilizado fue Oracle dado que, por su capacidad objeto relacional, permite la definición de dominios difusos mediante un enfoque orientado a objetos. Esta capacidad no es provista en su totalidad por el resto de los manejadores disponibles, en particular aquellos de uso libre. Oracle permite representar objetos complejos, extender tipos de datos nativos, variedad de métodos y es idónea para operar con otros lenguajes de programación además de SQL.

En el Modelo Objeto-Relacional que implementa Oracle se respetan las tres características propias de la programación orientada a objetos, a saber: a) *Encapsulación*, la clase contiene todos los atributos y métodos que son transparentes para el usuario; b) *Herencia*, una clase puede heredar atributos y métodos de otras; y c) *Polimorfismo*, una vez que se hereda de una clase, una sub-clase puede añadir/redefinir nuevos atributos y métodos.

En un manejador de bases de datos resulta muy útil para el programador tener esta funcionalidad tan completa y amplia ya que permite una modelación del universo de discurso mucho más fiel, completando los requerimientos de un proyecto de naturaleza compleja. La facilidad de crear un tipo de dato para ser utilizado incluso en procedimientos o funciones, es una herramienta muy poderosa a la hora de desarrollar una aplicación como la requerida ya que permite modelar los tipos de datos necesarios para crear la base de datos difusa sensible al contexto.

Luego de analizar los diferentes atributos del EFA que podrían tener valores imperfectos, la base de datos, diseñada en la sección anterior, fue extendida. Un diagrama de UML de clases que muestra el diseño conceptual de la Base de Datos para el Examen Físico Articular del Hospital Ortopédico Infantil de Venezuela se muestra en la figura 37, aquí la lista de atributos está limitada. El diccionario de datos que describe en detalle la lista de atributos para el Examen Físico Articular (EFA), exámenes especiales y articulaciones de miembros inferiores (cadera, rodilla, tobillo y pie) puede verse en el Apéndice B.

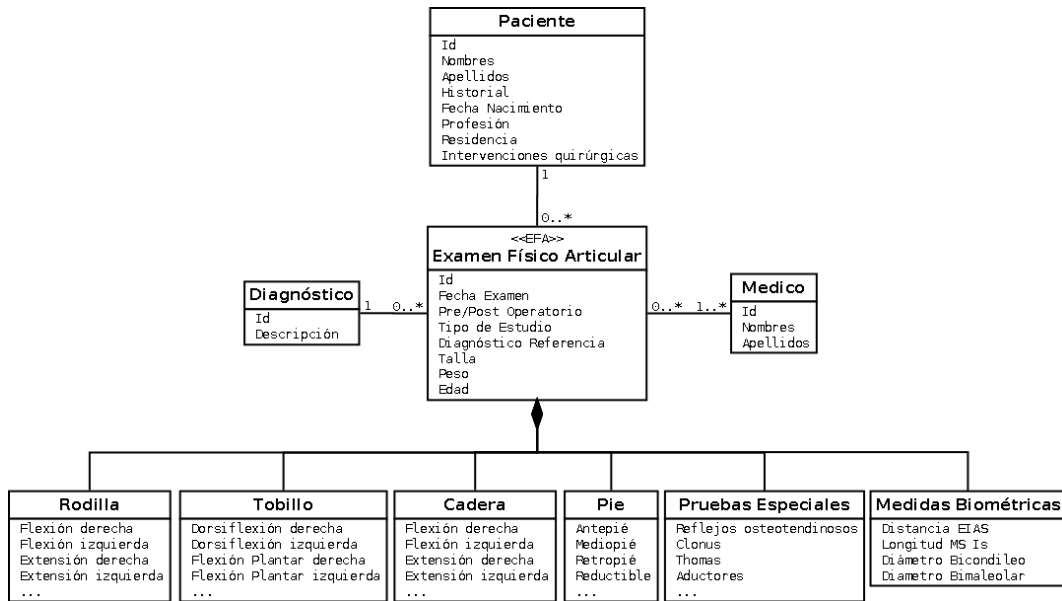


Figura 37. Diagrama de Clases del EFA

En la Figura 38 se muestra el diagrama de clases que presenta los dominios difusos extendidos para incorporarlos como nuevos tipos de datos en la base de datos del EFA.

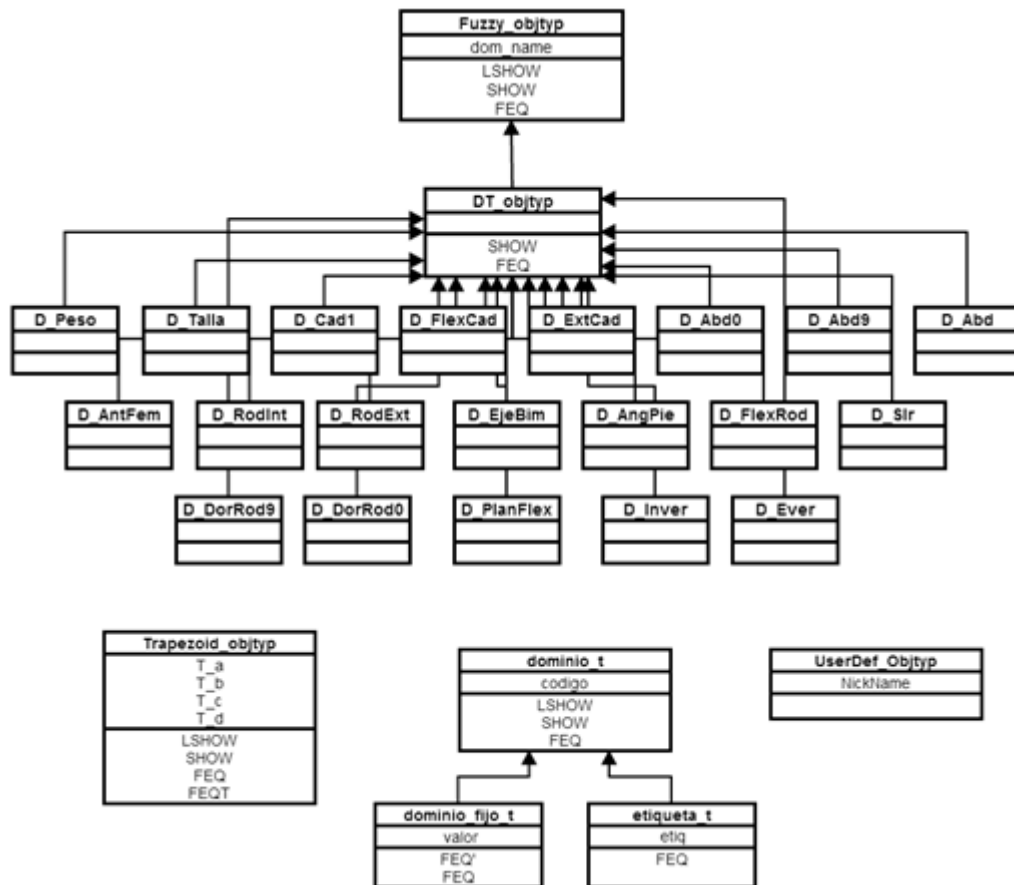


Figura 38. Diagrama de Clases Dominios Difusos

En la figura 39 se muestra un diagrama UML de componentes de la prueba de concepto desarrollada. Se observa las interfaces para clientes web y Móvil, además de los lenguajes utilizados antes mencionados. La aplicación para el Hospital Ortopédico Infantil del Examen Físico Articular (HOI-EFA), consta de cuatro módulos principales: Inicio Sesión-Registro, Ingreso de Datos, Edición de Datos y Consultas. En el *back-end* se encuentra la Base de Datos y los módulos CTP que sirven como mediadores entre la aplicación y el ingreso y recuperación de datos tal como se ha explicado durante esta tesis.

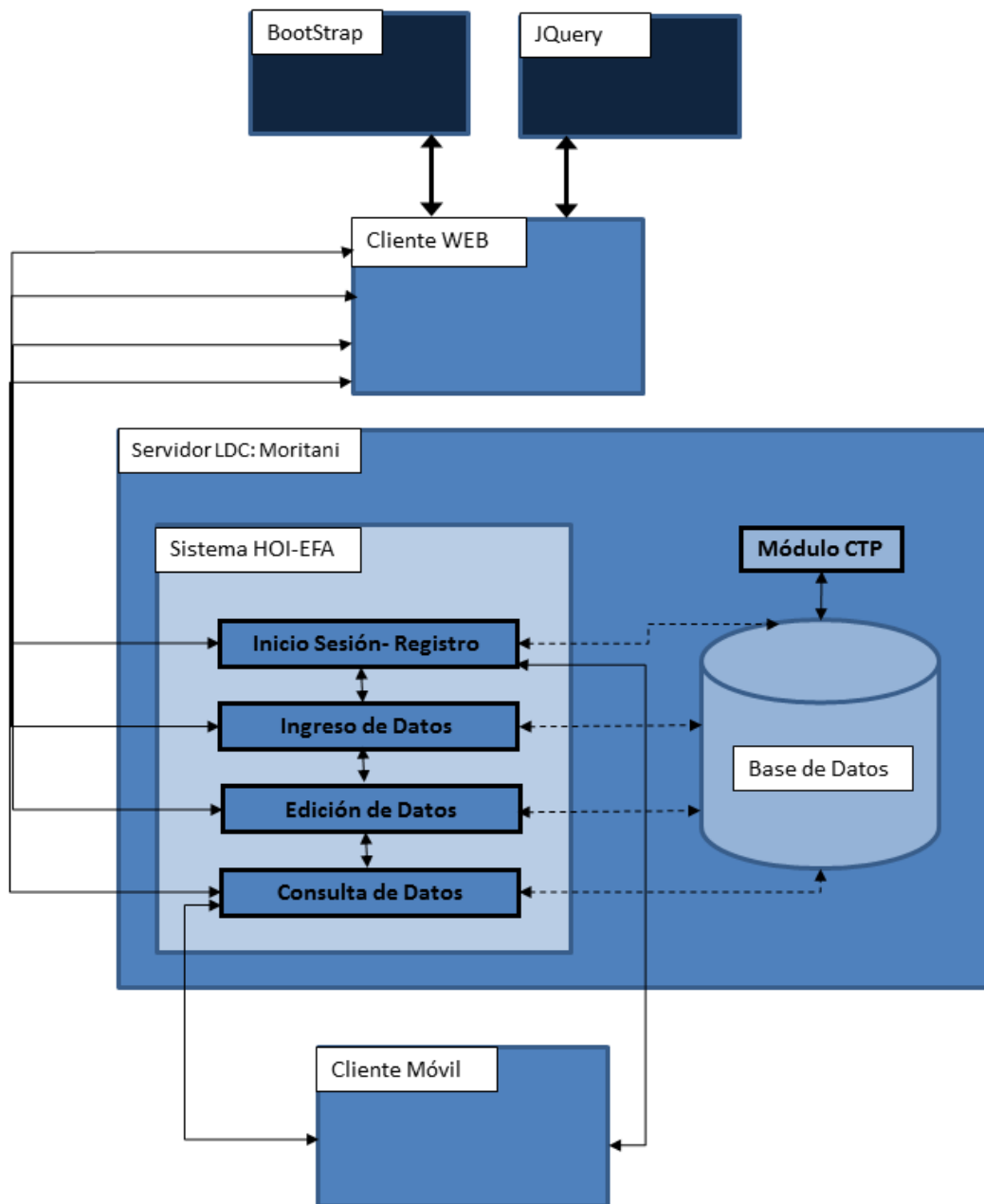


Figura 39. Diagrama de Componentes de la Aplicación

La clase *Fuzzy_objtyp* representa los dominios difusos. Esta clase posee los métodos LSHOW, SHOW y FEQ que implementan nuestro modelo propuesto para lograr la semántica adaptable. La clase *DT_objtyp* corresponde al dominio difuso atómico continuo. Las sub-clases que heredan de ésta se convertirán en nuevos tipos de datos a ser utilizado por todos los atributos con valores imperfectos que pueden ser representados por trapezoides (*Trapezoid_objtyp*) que tiene el examen físico articular, como por ejemplo, el peso y la talla.

Además se tiene la clase *dominio_t* la cual se utiliza para generalizar el dominio difuso atómico categórico (*etiqueta_t*) y el atómico discreto (*dominio_fijo_t*). Por último se define la clase *UserDef_Objtyp* que representa los usuarios que el sistema toma como defecto.

6.2.4. Interfaces de la Aplicación

A continuación se irán presentando las interfaces programadas que puede utilizar el usuario para interactuar en forma amigable tanto con la aplicación web como su versión para móviles con sistema operativo Android. En la figura 40 se presenta la página de inicio de sesión de la versión para web.

En el caso de que el usuario no esté registrado, debe hacerlo a través de la opción de registro (ver la figura 41), en otro caso el usuario se identifica con sus credenciales.

The image shows a web interface for a login page. At the top, there is a blue banner with the text "INICIO DE SESIÓN" in large white letters and "HOSPITAL ORTOPÉDICO INFANTIL" in smaller white letters below it. Below the banner, the text "INICIO DE SESIÓN" is repeated in a dark font. To the right, there are two input fields: "Usuario:" and "Contraseña:". Below the "Contraseña:" field is a black button with the text "INICIAR SESION" in white. At the bottom left, there is a black button with the text "REGISTRARSE" in white.

Figura 40. Página de Inicio de Sesión para la Aplicación Web

REGISTRO
HOSPITAL ORTOPÉDICO INFANTIL

DATOS PERSONALES DEL MEDICO

Nombres:

Apellidos:

Cédula de Identidad:

Usuario:

Contraseña:

Repetir Contraseña:

Tipo de Usuario:

Fisioterapeuta

Interpretador

Figura 41. Página de Registro

En el momento de registrarse además de suministrar sus datos personales el médico identificará su rol: Fisioterapeuta o Interpretador. Esta es una característica que se almacenará en su perfil y en el caso de que sea médico interpretador podrá hacer consultas, pero no registrar clientes ni resultados del examen físico articular. De esta forma el Rol sirve para que la aplicación sea sensible a las opciones que pueda presentar al tipo de usuario.

En la figura 42 se puede visualizar la pantalla de inicio si el usuario se conecta a través de un dispositivo móvil con sistema operativo Android. Al detectar este tipo de sistema operativo la aplicación es direccionada a la interfaz desarrollada mediante el *framework Bootstrap*. Así la interfaz se ajustará al tamaño de la pantalla del dispositivo (*responsive*).

Una vez registrado el usuario e identificado con un rol de Fisioterapeuta, entonces podrá ingresar sus parámetros, los cuales son: descripción de etiquetas para peso, matriz de semejanza para Tono Flexor Dorsal (derecho e izquierdo), matriz de semejanza para el tipo de marcha.

A efectos de hacer la prueba de concepto con diferentes tipos de dominios difusos se seleccionó el peso como un atributo que, aunque es preciso al momento de ingresar el valor obtenido en el examen físico, es una variable lingüística que puede ser descrita también por etiquetas (por ejemplo delgado, normal, obeso) cuya función de pertenencia es de tipo trapezoidal (dominio difuso atómico continuo).

El tono flexor dorsal es la capacidad de contracción del músculo del tobillo al acercar el pie derecho al plano longitudinal del tobillo con la rodilla a 90 grados de flexión y es medido en una escala de 1 al 4. Pero también se puede describir con las etiquetas lingüísticas atonía (falta de tono), hipotonía, normal e hipertonía (exceso de tono) con espasticidad. Así, este es un dominio difuso atómico discreto y se requiere de una matriz de semejanza (etiqueta - valor evaluado). En

la aplicación el usuario coloca la matriz en una hoja de cálculo que luego es cargada al sistema (tono flexor dorsal derecho y tono flexor dorsal izquierdo).



Figura 42. Inicio Interfaz Móvil

El tipo de marcha es un dominio difuso atómico categórico, ya que el evaluador observa la marcha del paciente y de acuerdo las características más resaltantes es colocada una etiqueta (normal, hemipléjica, atáxica, parkinsoniana, danzante, antiálgica y espástica). En este caso el usuario también coloca los datos de la matriz de semejanza en una hoja de cálculo que luego es cargada al sistema.

En la figura 43 se muestra cuando ya un usuario ha ingresado con un rol de fisioterapeuta lo cual es indicado por el sistema, como parte de sus datos personales en el perfil del usuario.

En la figura 44 se presenta los parámetros una vez que ya se han ingresado las etiquetas para identificar la variable lingüística peso. El médico al escoger la opción de consultar sus parámetros se le presenta gráficamente las funciones de pertenencia.

Después de haber ingresado a través de hojas de cálculos las matrices de pertenencia para tono flexor dorsal, se puede visualizar las matrices en el perfil de usuario, como se observa en la figura 45.



Figura 43. Datos Personales del médico



Figura 44. Etiquetas lingüísticas para el Peso



Figura 45. Matrices de semejanza para Tono Flexor Dorsal

Posteriormente el médico puede ingresar datos personales del paciente tales como: número de cédula de identidad, nombres, apellidos, profesión, lugar de residencia y fecha de nacimiento. Además, el médico debe llenar la ficha de historial donde se le indica el diagnóstico que hizo el médico referente, intervenciones quirúrgicas y dispositivos que utiliza el paciente para mejorar la marcha indicados por los médicos referentes (muleta, silla de rueda, andadera, bastón).

Los aditamentos utilizados es un ejemplo de un dominio difuso conjuntivo, porque un paciente puede utilizar varios a la vez. Al médico se le pide especificar con qué frecuencia los utiliza el paciente a través de etiquetas (siempre, frecuentemente, algunas veces o nunca) que el sistema traduce a grados de pertenencia. De esta forma luego se pueden comparar los aditamentos utilizados por los pacientes a través de una consulta.

Ya identificado el paciente con su historial e ingresado en la Base de Datos, el médico puede ingresar los datos correspondientes al Examen Físico Articular (EFA). En la figura 46 se muestra la interfaz propuesta para el ingreso de los datos del EFA en el modelo de bases de datos extendido.

INGRESO DE NUEVO EFA
HOSPITAL ORTOPÉDICO INFANTE

HOME NUEVO PACIENTE NUEVO EFA CONSULTAR PACIENTE ELIMINAR PACIENTE

General Cadera Rodilla Tobillo Pie Exam. Especiales

GENERAL

ID Historial: ID Persona: Estudio Pre/Post Operatorio: ID Laboratorio:

Fecha de Examen: Talla: Tipo de Estudio: Medico Interno:

Edad: Peso: Diagnostico de Referencia: Medico Fisioterapeuta:

(*) Estos campos son Obligatorios

Figura 46. Ingreso de un nuevo EFA

En la Figura 47 se pueden observar la interfaz móvil para ingresar datos del EFA. Es de hacer notar que por las restricciones del dispositivo, sólo se podrá ingresar ciertos datos del EFA, debido a que la interfaz móvil se utilizará preferentemente para hacer consultas. Los elementos de la interfaz móvil que se pueden visualizar en los íconos de la parte superior corresponden a: inicio (*home*), agregar paciente, agregar examen físico articular, consultar pacientes, eliminar registro de paciente y salir.

Datos Examen Físico Articular

Cedula:

Fecha:

Nombre Medico:

Peso (Kg.):

Seleccione:

Seleccione:

Seleccione:

Figura 47. Ingreso Datos EFA interfaz móvil

Para la adaptación del sistema web al ambiente móvil se tomaron en consideración diversos obstáculos que presentan los dispositivos móviles, tales como: tamaño de la pantalla, anchura y altura de variables, pantalla táctil, dificultad de escritura. Además, debido al ambiente físico desafiante que supone la movilidad es necesario considerar aspectos como la iluminación, el ruido o el movimiento. Se eligieron colores que contrastan para facilitar la visibilidad en la mayor parte de las situaciones en la que se localice el usuario.

Como una forma de presentar información imperfecta de forma amigable, se da la posibilidad al usuario de definir valores de la historia del paciente a través del uso etiquetas (Ver figura 48). Por ejemplo, en lugar de colocar el peso preciso de un paciente se puede seleccionar una etiqueta definida para dicha variable, entonces el sistema almacenará el trapecio correspondiente. Para el caso del tono flexor puede seleccionar una etiqueta (atonía, hipotonía, normotonía o hipertonía) y con una barra deslizadora se permite seleccionar un grado de pertenencia.

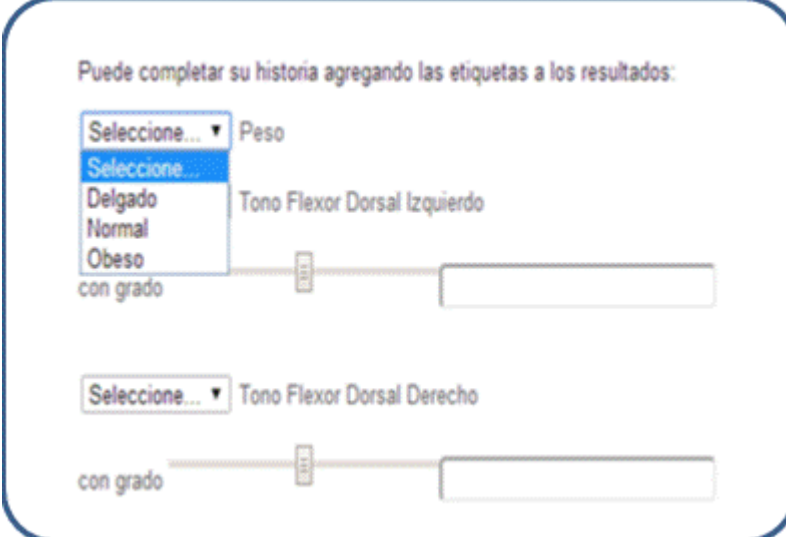


Figura 48. Ingreso de datos Imperfectos

6.2.5. Consultas

La aplicación de escritorio contempla la búsqueda de pacientes ingresando el número de cédula de identidad. En la interfaz móvil, dada la diversidad de lugares en los que el médico o fisioterapeuta pudiera encontrarse con poca información certera del paciente, se decidió aumentar esta funcionalidad agregando consultas por nombre o apellido. La información desplegada del paciente se organizó en varias pantallas para evitar desplazamientos innecesarios al moverse en ella y permitir la posibilidad de visualizarla en su totalidad. En la figura 49 se muestra el resultado de la búsqueda por nombre de la paciente “María Pérez”.

La consulta general de pacientes permite al usuario realizar consultas usando sus etiquetas lingüísticas, que han sido especificadas según sus preferencias. Estas preferencias fueron definidas previamente en la aplicación web. Por ejemplo, el médico puede solicitar los pacientes *delgados* con *marcha normal*, donde *delgado* y *normal* son etiquetas lingüísticas definidas por éste. Si el usuario no ha colocado sus preferencias se toman los valores por defecto, los cuales pueden ser modificados cuando el usuario desee. El resultado obtenido depende de las definiciones (ver Figura 50).

Se da la posibilidad de consultas que retornen pacientes comparando aditamentos de apoyo a su movilidad. Debido a las limitaciones que presenta la pantalla de un dispositivo móvil se provee una respuesta con un máximo de ocho pacientes los cuales son ordenados por el grado de semejanza, de manera decreciente. Para esto se utiliza un “umbral” (*threshold*) o valor en el intervalo $[0,1)$, definido por el usuario, que filtra los resultados con un grado de semejanza menor al deseado. De esta manera, se evita la sobrecarga de información poco relevante.

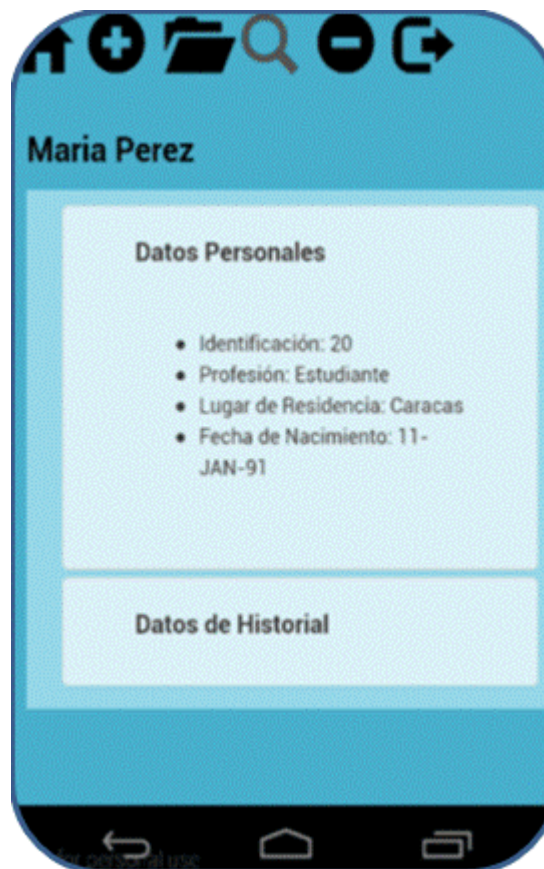


Figura 49. Datos de un Paciente



The screenshot shows a mobile application interface with a blue header. At the top, there is a navigation bar with icons for home, add, folder, search, minus, and refresh. Below the navigation bar, the text 'Fisioterapeuta: Doctor Who' is displayed. The main title is 'Resultados'. Below the title is a table with the following data:

Cédula	Fecha Examen	Peso	Características de Marcha
30	12-FEB-12	.7	.4
22	24-JAN-14	.7	.4
24	01-JAN-14	1	.4

Figura 50. Resultado de la Consulta de Pacientes *Delgados* con Marcha *Normal*

En la Figura 51 se puede ver el resultado de una consulta que compara a los aditamentos de movilidad de María Pérez con los demás pacientes cuyo grado de semejanza es mayor a 0.3. Aquí se está sensibilizando el tamaño del resultado además de utilizar una medida de orden como son los grados de pertenencia para dar sólo las respuestas más relevantes al usuario.



The screenshot shows a mobile application interface with a blue header. At the top, there is a navigation bar with icons for home, add, folder, search, minus, and refresh. Below the navigation bar, the text 'Fisioterapeuta: Doctor Who' is displayed. The main title is 'Resultados de Maria Perez'. Below the title is a table with the following data:

Nombres	Apellidos	Cedula	Semejanza	Prom.Semej
Maria	Perez	20	1	1
Ana	Ramirez	26	.75	.75
Elena	Barreto	32	.75	.88
Gabriel	Marcano	23	.75	.88
Tatiana	Murillo	34	.75	.75

Below the table, the text 'Umbral: 0.3' is displayed.

Figura 51. Resultado de una Comparación entre Pacientes

Como ejemplo de consultas con atributos que pueden ser precisos, pero al ser variables lingüísticas pueden ser representados por etiquetas lingüísticas, las cuales pueden ser utilizadas a nivel de las consultas, tenemos en nuestra

aplicación del peso de una persona. Cada usuario puede definir un conjunto de etiquetas para el peso de una persona. Así, una vez efectuado el examen físico articular, a pesar de que se almacena en la base de datos un dato que puede ser preciso (por ejemplo 85 Kg) las consultas pueden ser difusas de acuerdo a las etiquetas definidas.

En la Figura 52 se muestra el resultado de la consulta “pacientes con peso *normal*” efectuada por el usuario “Luis”, mientras que en la Figura 53 se muestra el resultado de la misma consulta efectuada por “Rubén”, la columna peso es el grado de pertenencia a la etiqueta “normal” definida por cada usuario.

En el primer caso la paciente “Andreina Lorienté” aparece en el resultado de la consulta con un grado de pertenencia de 0.5 y el paciente “Andrés Lorienté” con un valor de pertenencia de 0.33. Son los pacientes cuyo peso igual a *normal* tienen un valor de pertenencia distinto de cero.

En el segundo caso, también aparece “Andreina Lorienté”, pero con un grado de pertenencia igual a 1, mientras que “Andrés Lorienté” no aparece en el resultado de la consulta, esto indica que su valor de pertenencia es cero. En su lugar, aparece otro paciente “Verónica Hernández” con grado de pertenencia 0.33

Nombres	Apellidos	Cedula	Id Historial	Fecha Examen	Peso
andreina	loriente	19380899	1	11-20-2012	.5
andres	loriente	19380900	3	11-20-2012	.33

Figura 52. Resultado de la Consulta “Pacientes con Peso *Normal*” para “Luis”

Nombres	Apellidos	Cedula	Id Historial	Fecha Examen	Peso
andreina	loriente	19380899	1	11-20-2012	1
veronica	hernandez	19380898	2	11-30-2012	.33

Figura 53. Resultado de la Consulta “Pacientes con Peso *Normal*” para “Rubén”

En estas figuras, se puede notar claramente que los grados de pertenencia son distintos aunque se están accediendo a los mismos datos. Debido al hecho que cada doctor puede tener una interpretación diferente del peso normal de una persona, los resultados son distintos.

Otro ejemplo de la sensibilización al contexto es cuando el usuario puede realizar consultas por diferentes atributos difusos utilizando su propia semántica.

En el resultado aparecerán las etiquetas definidas por los usuarios acompañadas del grado de pertenencia del valor del atributo a dicha etiqueta (conjunto difuso). Esto puede observarse en las figuras 54 y 55.

CI	Nombre	Fecha	Peso	Talla
19254844	Jhosbert Abraham	10-jun-14	0/Flaco; 0/Gordo 1/Sano;	0/Bajo; 0/Alto ,33/Normal;
19254844	Jhosbert Abraham	08-jun-14	0/Flaco; 0/Gordo 1/Sano;	0/Bajo; 0/Alto ,33/Normal;
20652525	Andel	08-jun-14	,44/Flaco; 0/Gordo 0/Sano;	,6/Bajo; 0/Alto ,33/Normal;
6368339	Jhon W	08-jun-14	0/Flaco; 0/Gordo ,53/Sano;	0/Bajo; 1/Alto 0/Normal;
81291961	Carlos	18-jun-14	0/Flaco; 0/Gordo 1/Sano;	0/Bajo; ,67/Alto 0/Normal;
3666968	Nancy	18-jun-14	0/Flaco; 0/Gordo 1/Sano;	,6/Bajo; 0/Alto 0/Normal;
9517539	Alejandro	20-jun-14	0/Flaco; ,2/Gordo 0/Sano;	0/Bajo; ,67/Alto 0/Normal;
19509671	Maria Isabel	12-jun-14	,2/Flaco; 0/Gordo 0/Sano;	0/Bajo; ,67/Alto 0/Normal;
5528577	Aleida	18-jun-14	0/Flaco; 0/Gordo ,17/Sano;	0/Bajo; 0/Alto 1/Normal;
6271739	Emiro	18-jun-14	0/Flaco; ,5/Gordo 0/Sano;	0/Bajo; ,67/Alto 1/Normal;
6271739	Emiro	08-feb-14	0/Flaco; 1/Gordo 0/Sano;	0/Bajo; ,67/Alto 1/Normal;
20652525	Andel	05-DEC-12	1/Flaco; 0/Gordo 0/Sano;	,6/Bajo; 0/Alto ,33/Normal;
3666968	Nancy	11-DEC-99	1/Flaco; 0/Gordo 0/Sano;	0/Bajo; 0/Alto ,8/Normal;
19509671	Maria Isabel	01-feb-00	1/Flaco; 0/Gordo 0/Sano;	0/Bajo; 0/Alto ,8/Normal;
81291961	Carlos	25-APR-13	0/Flaco; 0/Gordo 1/Sano;	0/Bajo; ,5/Alto 1/Normal;
9517539	Alejandro	16-nov-12	0/Flaco; 0/Gordo 1/Sano;	0/Bajo; ,4/Alto 1/Normal;

Figura 54. Consulta Difusa del Examen Físico Articular con semántica de Peso y Talla del “Luis”

CI	Nombre	Fecha	Peso	Talla
19254844	Jhosbert Abraham	10-jun-14	0/Delgado; 1/Normal 0/Obeso;	0/Bajo; 0/Alto ,8/Normal;
19254844	Jhosbert Abraham	08-jun-14	0/Delgado; 1/Normal 0/Obeso;	0/Bajo; 0/Alto ,8/Normal;
20652525	Andel	08-jun-14	,07/Delgado; ,08/Normal 0/Obeso;	,8/Bajo; 0/Alto ,2/Normal;
6368339	Jhon W	08-jun-14	0/Delgado; ,2/Normal ,08/Obeso;	0/Bajo; 1/Alto 0/Normal;
81291961	Carlos	18-jun-14	0/Delgado; ,67/Normal 0/Obeso;	0/Bajo; ,7/Alto 0/Normal;
3666968	Nancy	18-jun-14	0/Delgado; ,67/Normal 0/Obeso;	,4/Bajo; 0/Alto 0/Normal;
9517539	Alejandro	20-jun-14	0/Delgado; 0/Normal ,56/Obeso;	0/Bajo; ,8/Alto 0/Normal;
19509671	Maria Isabel	12-jun-14	0/Delgado; ,58/Normal 0/Obeso;	0/Bajo; ,6/Alto 4/Normal;
5528577	Aleida	18-jun-14	0/Delgado; 1/Normal 0/Obeso;	0/Bajo; 0/Alto 1/Normal;
6271739	Emiro	18-jun-14	0/Delgado; 0/Normal ,8/Obeso;	0/Bajo; 0/Alto 1/Normal;
6271739	Emiro	08-feb-14	0/Delgado; 0/Normal 1/Obeso;	0/Bajo; 0/Alto 1/Normal;
20652525	Andel	05-DEC-12	1/Delgado; 0/Normal 0/Obeso;	,8/Bajo; 0/Alto ,2/Normal;
3666968	Nancy	11-DEC-99	1/Delgado; 0/Normal 0/Obeso;	0/Bajo; 0/Alto ,1/Normal;
19509671	Maria Isabel	01-feb-00	1/Delgado; 0/Normal 0/Obeso;	0/Bajo; 0/Alto ,1/Normal;
81291961	Carlos	25-APR-13	0/Delgado; 1/Normal 0/Obeso;	0/Bajo; ,8/Alto 0/Normal;
9517539	Alejandro	16-nov-12	0/Delgado; ,8/Normal 0/Obeso;	0/Bajo; ,6/Alto 0/Normal;

Figura 55. Consulta Difusa del Examen Físico Articular con semántica de Peso y Talla del “Rubén”

Allí se muestran todos los EFA realizados a los pacientes en diferentes fechas, y los valores del peso y de la talla representados como conjuntos difusos. Cada etiqueta definida por el usuario viene acompañada de su grado de pertenencia.

Así por ejemplo, en la Figura 54, para el paciente “Jhosbert Abraham”, según la definición del “Luis” tiene un peso *Sano* con grado 1 y talla *Normal* con

grado 0.33 en el EFA realizado el “10-jun-14”, mientras que el paciente “Andel” tiene peso *Flaco* con grado 0.44 y talla *Bajo* con grado 0.6.

Si observamos la Figura 55, para el mismo paciente “Jhosbert Abraham”, en la misma fecha “10-jun-14”, el resultado obtenido es diferente pues la definición del “Rubén” considera otros valores y etiquetas. Este usuario define para el peso las etiquetas *Delgado*, *Normal* y *Obeso*, en lugar de *Flaco*, *Gordo* y *Sano*. Además, sus definiciones de los trapecios para estas etiquetas son diferentes pues los resultados no son los mismos. Para “Rubén”, tiene un peso *Normal* con grado 1 y talla *Normal* con grado 0.8, mientras que el paciente “Andel” tiene peso *Delgado* con grado 0.7 y *Normal* con grado 0.8, y talla *Bajo* con grado 0.8 y *Normal* con grado 0.2. Es aquí donde se refleja que los usuarios usan semánticas diferentes para términos descriptivos de la misma variable lingüística.

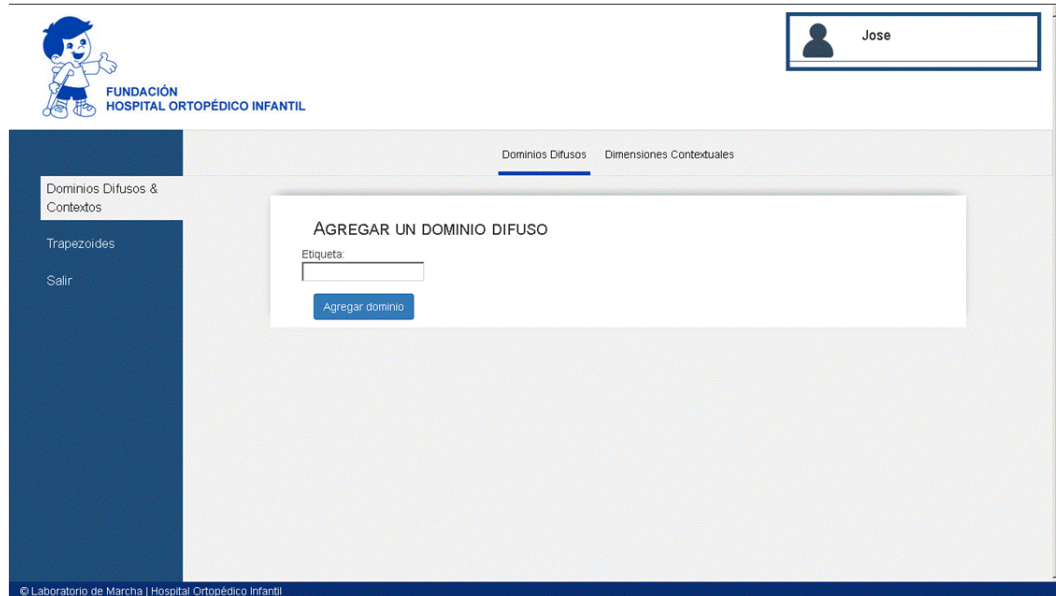
La sensibilización al contexto se nota en que aunque la base de datos de EFA de pacientes es la misma, los conjuntos difusos obtenidos para el peso y la talla son diferentes. Esto se debe a las diferentes definiciones de etiquetas para las mismas variables lingüísticas del “Luis” con respecto al “Rubén”. Por lo que al cambiar el usuario se cambia el contexto, y el sistema reconoce los diferentes lenguajes produciendo una salida personalizada de acuerdo a las definiciones de cada usuario.

6.2.6. Sesión de Administrador

En la prueba de concepto también hemos diseñado una sesión para que el administrador del Sistema pueda ingresar datos correspondientes a dominios difusos, dimensiones contextuales, dependencias contextuales, dominios contextuales y etiquetas por defecto, de esta forma se pueden ingresar amigablemente datos del catálogo contextual, sin necesidad de hacerlo a través de instrucciones de SQL. En la figura 56 se presenta la interfaz para introducir un dominio difuso.

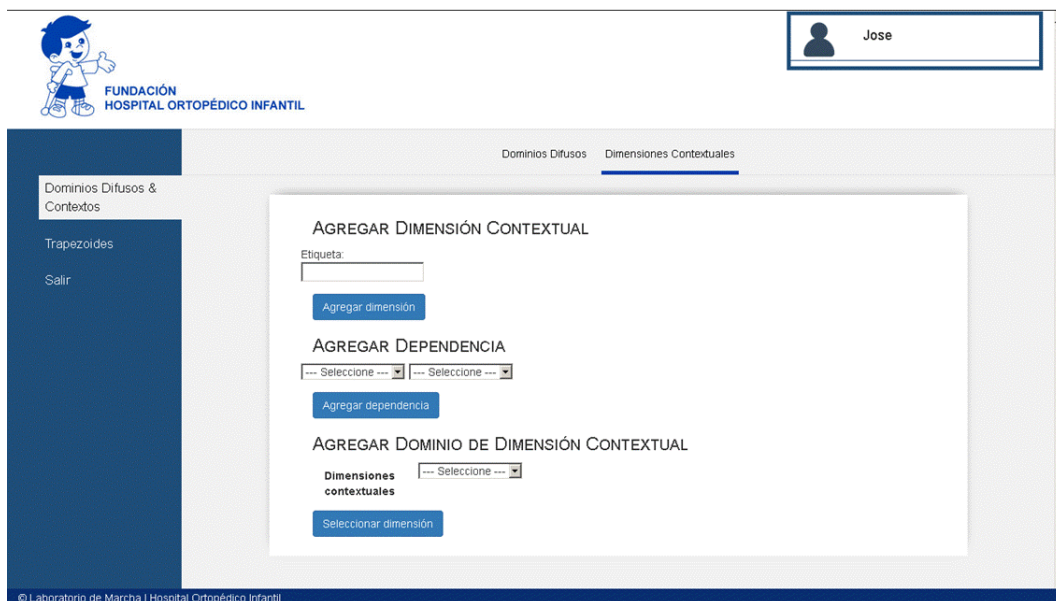
Por otra parte a través de la interfaz mostrada en la figura 57 se pueden ingresar dimensiones contextuales, establecer dependencias contextuales estableciendo un dominio difuso y una dimensión contextual. Además se pueden agregar dominios de esta última.

Finalmente se propone agregar trapecoides por defecto para el contexto actual a través de la interfaz mostrada en la Figura 58. El usuario escoge un dominio difuso, luego se coloca la etiqueta lingüística y los datos del trapecio colocando los 4 puntos de inflexión del mismo. Éste trapecio representará a la etiqueta por defecto en el contexto actual en que se ingresa el mismo.



The screenshot shows a web application interface for the 'FUNDACIÓN HOSPITAL ORTOPÉDICO INFANTIL'. The user is logged in as 'Jose'. The main menu on the left includes 'Dominios Difusos & Contextos', 'Trapezoides', and 'Salir'. The top navigation bar has 'Dominios Difusos' and 'Dimensiones Contextuales'. The central panel is titled 'AGREGAR UN DOMINIO DIFUSO' and contains a form with an 'Etiqueta:' label, an input field, and an 'Agregar dominio' button. The footer indicates '© Laboratorio de Marcha | Hospital Ortopédico Infantil'.

Figura 56. Interfaz para ingreso de dominios difusos



The screenshot shows the same web application interface, but the user is now in the 'Dimensiones Contextuales' section. The main panel is titled 'AGREGAR DIMENSIÓN CONTEXTUAL' and contains three sections: 1) 'AGREGAR DIMENSIÓN CONTEXTUAL' with an 'Etiqueta:' label, an input field, and an 'Agregar dimensión' button. 2) 'AGREGAR DEPENDENCIA' with two dropdown menus labeled '--- Seleccione ---' and an 'Agregar dependencia' button. 3) 'AGREGAR DOMINIO DE DIMENSIÓN CONTEXTUAL' with a dropdown menu labeled 'Dimensiones contextuales' and '--- Seleccione ---' and a 'Seleccionar dimensión' button. The footer remains the same: '© Laboratorio de Marcha | Hospital Ortopédico Infantil'.

Figura 57. Interfaz para ingreso de dimensiones contextuales

En la siguiente sección se describe cómo se aplicó la arquitectura propuesta en el capítulo 5 de SBDDSC en la aplicación real presentada con la implementación en las secciones anteriores.

The screenshot shows a web application interface. At the top left is the logo for 'FUNDACIÓN HOSPITAL ORTOPÉDICO INFANTIL'. At the top right is a user profile for 'Jose'. The main content area is titled 'Trapezoides por defecto' and contains a form titled 'AGREGAR UN TRAPEZOIDE POR DEFECTO'. The form has a dropdown menu for 'Dominio', a text input for 'Etiqueta', and five text inputs for 'Limite izquierdo de soporte', 'Limite izquierdo de núcleo', 'Limite derecho de núcleo', and 'Limite derecho de soporte'. A 'Seleccionar' button is at the bottom left of the form. The footer of the page reads '© Laboratorio de Marmas | Hospital Ortopédico Infantil'.

Figura 58. Interfaz para ingreso de trapezoide por defecto

6.3. Utilización de la Arquitectura Propuesta en la Aplicación EFA-HOI

En las secciones anteriores hemos presentado una implementación en SQL y una aplicación web con interfaz móvil que denominamos EFA-HOI (Examen Físico Articular del Hospital Ortopédico Infantil). A continuación describimos cómo fue utilizado el marco que proporciona la arquitectura propuesta para el concepto de prueba desarrollado en el área médica. En la Figura 59 se muestra los módulos utilizados para el desarrollo de Sistemas de Bases de datos Difusas Sensibles al Contexto (SBDDSC).

A través de la implementación en SQL los usuarios pueden interactuar directamente con los módulos CTP para el almacenamiento de datos (CAS) sensibles al contexto a través de cláusulas INSERT de SQL; o para la recuperación de información (CAR) sensibles al contexto mediante la utilización de consultas de SQL. Estos módulos se encargan de hacer las transformaciones lingüísticas necesarias para almacenar los datos utilizando los dominios difusos predefinidos por el diseñador del sistema.

Por otro lado, usuarios inexpertos o administradores del sistema podrán interactuar con el SBDDSC a través de aplicaciones desarrolladas para un propósito específico tal como la mostrada en la sección anterior. El usuario interactúa con interfaces web para el ingreso de datos la cual utiliza la capa intermedia (módulos CTP) para que los datos almacenados sean sensibles al contexto. Además la recuperación de información sensible al contexto es

efectuada a través de la aplicación web o su interfaz móvil quienes se comunican con el sistema utilizando el módulo CAR:

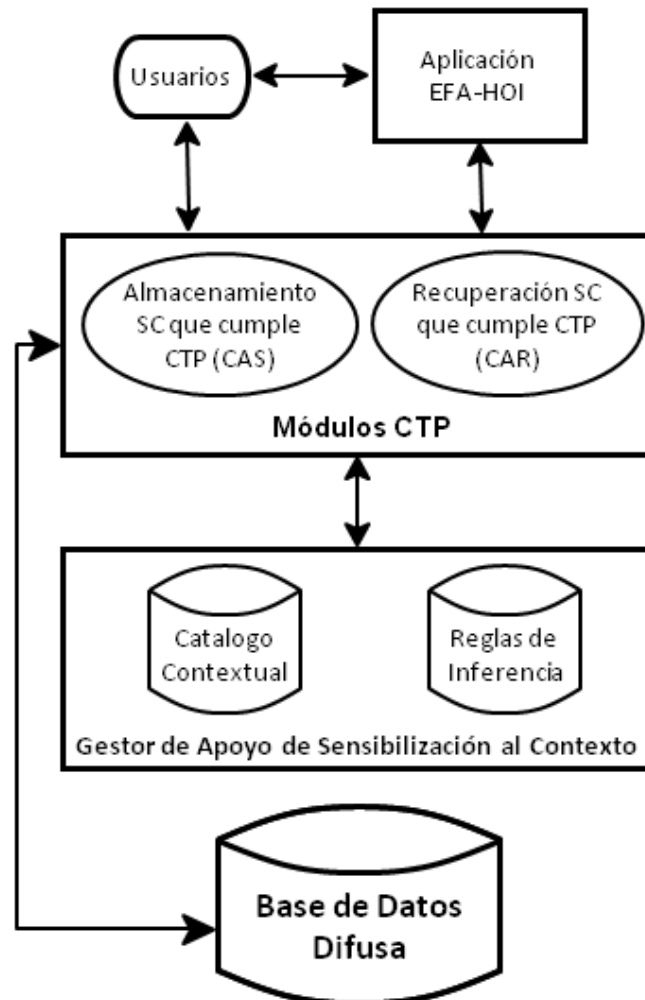


Figura 59. Utilización de la Arquitectura Propuesta en la prueba de concepto

El Gestor de Apoyo de Sensibilización al Contexto (CASM) contiene el catálogo contextual representado principalmente por la información que asocia las etiquetas lingüísticas con los dominios difusos y su contexto. Las reglas de inferencia programadas, utilizando el estándar SQL:99, se implementan con métodos y procedimientos (o funciones) almacenadas, los cuales pueden ser utilizados directamente en una línea de comandos o administrador de SQL, o ser llamadas desde la aplicación web.

No hemos olvidado que nuestra propuesta puede ser utilizada para múltiples dimensiones contextuales. En este caso en el catálogo contextual debe ser almacenado el contexto como una lista de pares 2-tupla <dimensión contextual, dominio contextual>, verificando las dependencias contextuales definidas para cada dominio difuso. El contexto actual entonces debe ser

verificado para que coincida con los valores de la lista almacenada en el contexto actual y en el caso de coincidir utilizar la semántica definida en el catálogo. Además debe tomarse en cuenta el parámetro “Always” si es definido sólo para un contexto específico, para todos los contextos del usuario o para cualquier contexto.

Capítulo 7

Conclusiones y Trabajos Futuros

Para finalizar la presente memoria se desarrolla la sección de las conclusiones obtenidas de esta investigación, para describir posteriormente los trabajos futuros que se pueden derivar de la misma.

7.1. Conclusiones

Las aplicaciones deben ser sensibles al contexto para que las Tecnologías de Información y Comunicación se adapten al usuario permitiendo que sean amigables para las personas, contribuyendo así a la mejora de la calidad de vida de las mismas.

El principal énfasis de los sistemas sensibles al contexto desarrollados hasta ahora está muy atado a datos que pueden obtener sensores de ambiente, por lo que se requieren nuevos enfoques para abordar el problema del contexto prestando mayor importancia a aspectos humanos tales como las preferencias, objetivos y estado del usuario mediante aplicaciones inteligentes.

Luego de identificar la importancia del contexto como factor determinante para desarrollar Sistemas de Información inteligentes, flexibles y personalizados; dado que muy poco se ha investigado en el área de bases de datos, decidimos abordar este difícil problema.

Aunado a ello, nos interesó el estudio de procesos de naturaleza humana como la comunicación entre personas y el lenguaje natural utilizado, que es ambiguo, subjetivo y altamente dependiente del contexto. En base a nuestro conocimiento sobre el uso de la lógica difusa para representar más fielmente este tipo de procesos, con ventajas documentadas sobre sistemas tradicionales, decidimos efectuar un estudio interdisciplinar de varias áreas de computación para presentar un modelo formal para el desarrollo de Sistemas de Bases de Datos Difusas Sensibles al Contexto (SBDDSC).

A manera de resumir las contribuciones de esta investigación podemos puntualizar los siguientes aportes que se pueden revisar a lo largo de los últimos tres capítulos de esta tesis.

- En la sección 4.1 se ha definido un esquema de interacción entre un usuario con un Sistema de Base de Datos Sensible al Contexto (SBDSC). Esto permite enfocarnos en el problema de estudio.
- En la sección 4.2 se ha propuesto un modelo de contexto tomando en cuenta dicho esquema de interacción; hemos procurado un modelo genérico pero también flexible para que sea de utilidad para un dominio específico de aplicación. El modelo de contexto propuesto considera dimensiones relacionadas tanto con el usuario como con el ambiente, dejando abierta la posibilidad de definir otras dimensiones en base al dominio concreto de aplicación.
- En la sección 4.3 hemos analizado los aspectos que se deben tener en cuenta en cuanto a la sensibilidad al contexto en la salida de un SBDSC, abordando el problema de definir en qué forma pueden verse afectados los resultados de la interacción con el usuario tomando en cuenta el modelo de contexto que establecimos para estos sistemas.
- En la sección 4.4 hemos propuesto una Arquitectura para Sistemas de Bases de Datos Sensibles al Contexto que sirve como marco de trabajo para desarrollar este tipo de sistemas en base al esquema de interacción analizado en apartados previos de la memoria. Éste es uno de los aportes principales de nuestro trabajo, ya que fija un marco general para el desarrollo de Sistemas de Información personalizados, basado en el modelo de contexto y esquema de interacción definidos. La arquitectura propuesta desarrolla las ideas de Pitoura, Stefanidis y Zaslavsky [116] para integrar el contexto en un Sistema Gestor de Base de Datos (SGBD).
- En el Capítulo 5 se aporta un modelo formal que permite la representación y uso de atributos lingüísticos con semántica adaptable al contexto en una base de datos. Esta propuesta se basa en el uso de conceptos propios de las bases de datos difusas, que forman la capa física de nuestro modelo. Se utilizó como fundamento de nuestra propuesta el marco teórico postulado en sus diversas investigaciones por Zadeh, propicio para adecuar las percepciones humanas a aplicaciones informáticas, mediante la utilización de lógica y conjuntos difusos, distribuciones de posibilidad, variables y etiquetas lingüísticas, entre otros. Esto permite la representación de datos imperfectos y el enriquecimiento semántico en las consultas a base de datos, flexibilizándolo mediante una particularización del modelo al desarrollar un marco teórico para los atributos lingüísticos con semántica adaptable al contexto. El modelo formal propuesto gira en

torno a los conceptos fundamentales de dimensión contextual y dependencia contextual, que permiten hacer convivir de manera armónica y dinámica en la base de datos las visiones del diseñador y de los distintos usuarios y sus contextos.

- También en el Capítulo 5, se aporta una implementación transparente de la propuesta formal anterior sobre un SGBDOR comercial como Oracle, al que se han incorporado capacidades de manejo de datos difusos y una lógica que desarrolla los elementos apropiados de la arquitectura de SBDDSC para hacer posible el uso amigable de atributos lingüísticos en dicho sistema. El diseño de instrucciones incorporando nuestro modelo formal permite la interacción con las nuevas funcionalidades a través de SQL:99.
- En la sección 6.1 se describe una implementación de una aplicación real utilizando el Sistema Gestor de Base de Datos Objeto-Relacional Oracle. Para ello se extienden los tipos de datos para representar dominios difusos, se definen métodos y procedimiento almacenados, además de mostrar la utilización de sentencias estándar de SQL para el ingreso y recuperación de datos en un Sistema de Base de Datos Difusa Sensible al Contexto (SBDDSC).
- Finalmente en la sección 6.2 se describe una prueba de concepto de una aplicación web con su interfaz móvil de un SBDDSC. Se presentan interfaces para la aplicación, consultas de muestra y el diseño de una sesión para el administrador del sistema. El uso de un dominio real de aplicación pone de manifiesto la utilidad de los modelos e implementaciones desarrolladas en el marco de este trabajo de investigación, a través de un prototipo de prueba de concepto que ha sido desarrollado con la colaboración de personal del Hospital Ortopédico Infantil de Venezuela.

Otras conclusiones a las que llegamos luego de la revisión de antecedentes y análisis documental, se pueden resumir en:

- La importancia de la perspectiva humana en los diversos sistemas informáticos de hoy en día, donde está involucrado el contexto del usuario en una era de grandes volúmenes de datos. Debido a esto es primordial el desarrollo de estudios con un enfoque interdisciplinar, tal como se planteó en esta investigación, que permitan a las personas mejorar el acceso a información de calidad a los usuarios por las diversas aplicaciones.
- El modelo de contexto debe estar centrado en un sub-problema específico para su aplicabilidad práctica y usabilidad. Es así como nos enfocamos en

nuestra propuesta en el esquema de interacción de un usuario con un Sistema de Base de Datos.

- Se aprovecharon experiencias sobre el contexto en Interacción Persona-Ordenador, Computación Ubicua, Inteligencia Artificial y Sistemas de Información. Entre las cuales destacan: la utilización del perfil de usuario, un modelo de capas para abordar la complejidad en la gestión y la adquisición de información contextual de alto nivel.
- Varias investigaciones de bases de datos se han enfocado en el tema de las preferencias contextuales, donde cada usuario en particular debe escoger sus preferencias en hipotéticos contextos. Se han hecho propuestas también para inferir estas preferencias contextuales. Nosotros proponemos no agobiar al usuario añadiendo complejidad a los sistemas de información que algunas veces no sea necesaria; nos enfocamos más bien en utilizar datos del contexto actual, estudiando cómo afectan éstos en la interacción entre el usuario y la base de datos, con el objetivo de mejorar la comunicación entre el usuario y el sistema, eliminando respuestas que pueden ser consideradas fuera de contexto o información ruido.
- Estudiamos técnicas de modelado orientado a objetos, así como la implementación de bases de datos difusas en Gestor de Bases de datos. Los antecedentes en cuanto a investigaciones de Bases de Datos utilizando modelos orientados a objeto más significativos estudiados fueron el trabajo de Vila et al. [154], además de Marín [90]; los cuales sirvieron de base fundamentales para el diseño del modelo propuesto.
- También se deben destacar los aportes sobre la implementación en gestores de base de datos objeto relacional de Cuevas et al. [42] y Cuevas [41], así como Barranco [12] que sirvieron de guía para llevar a cabo la implementación del diseño bajo Oracle. Dadas las facilidades de este SGBDOR se puede afirmar, de acuerdo al alcance de nuestro conocimiento, que es factible realizar una propuesta de semántica adaptable de acuerdo al contexto del usuario en aplicaciones de bases de datos difusas, cumpliendo con los postulados de *soft computing*: llevar a cabo una solución tratable, robusta, de bajo costo y lo más cercano a la realidad posible, utilizando aplicaciones inteligentes.

Específicamente en cuanto a la propuesta de un Modelo de Bases de Datos Sensible al Contexto, se tiene:

- La propuesta es una contribución a sistemas sensibles al contexto de alto nivel donde utilizamos herramientas poderosas que pueden potenciar este tipo de aplicaciones, a saber: modelo de datos orientado a objeto, Bases de Datos Difusas y Sistemas Gestores de Bases de Datos Objeto-Relacional.

- Los usuarios serán los más beneficiados al interactuar con sistemas personalizados que capturan las percepciones del mundo real lo más cercano posible, representan datos imperfectos, permitan hacer consultas basadas en sus preferencias tomando en cuenta datos contextuales, logrando así la mejora en la calidad del acceso a la información mediante aplicaciones inteligentes.
- A través del diseño del modelo contextual orientado a objetos con dominios difusos, se demostró que es directo implementarlo en un SGBDOR, tal como Oracle, mostrando instrucciones en forma general, y particular en la prueba de concepto, para su utilización bajo el estándar de SQL:99.

Seguidamente se abordan los posibles trabajos futuros donde se puede profundizar y abrir nuevas perspectivas en el área de Sistemas de Bases de Datos Difusas Sensibles al Contexto.

7.2. Trabajos Futuros

Dado lo novedoso de la propuesta se abren una serie de líneas de investigación partiendo de este estudio, entre las que se pueden mencionar:

- Proponer un modelo contextual orientado a objetos que gestione incertidumbre a través de lógica y conjuntos difusos, que pueda ser utilizado tanto en el ámbito de aplicaciones tal como Bases de Datos Difusas o en aplicaciones de otras áreas (Computación Ubicua, Inteligencia Artificial, Sistemas de Recuperación y Recomendación). Es decir, la información contextual en sí es una información que puede estar afectada de imprecisión e incertidumbre (en función de las fuentes que la producen, por ejemplo). Si bien una representación *sencilla* de dicha información contextual imperfecta es *bastante directa* cuando se usa como fundamento una base de datos difusa (que es el caso de esta memoria), su manipulación a través de dependencias requiere de desarrollos teóricos adicionales que esperamos abordar en el futuro a través de un modelo multidimensional jerárquico de contexto.
- Continuar con el desarrollo de interfaces amigables de aplicaciones de gestión sensibles al contexto, en otros dominios de aplicación, para dar un mayor poder expresivo a los sistemas de información, basado en los postulados de Computación ubicua, Interacción Persona-Ordenador y las teorías de Zadeh, para darle significado al contexto del usuario mediante aplicaciones inteligentes, aprovechando herramientas

poderosas como Modelos Orientado a Objetos, Base de Datos Difusas y Gestores de Base de Datos Objeto-Relacional.

- Incluir la capacidad de trasladar la información de aplicaciones sensibles al contexto a sistemas que manejan datos semi-estructurados (XML), de gran importancia hoy en día para poder interoperar con infraestructuras de servicios y aplicaciones Web. Los sistemas de datos semi-estructurados son el eje de áreas de gran actualidad en el ámbito de los Sistemas de Información, como el área del Big Data, que utiliza modelos de representación basados en grafos para el manejo de ingentes cantidades de datos que, por su forma, no encajan con facilidad en sistemas estructurados como el relacional o los basados en objetos que son la base de este trabajo de investigación. Sin embargo, las propuestas desarrolladas en esta memoria son trasladables a otros modelos de datos como los mencionados basados en grafos.
- Extender la semántica adaptable a ambientes de Computación Ubicua; estudiando la posibilidad de que los datos provenientes de sensores pueden estar representados por otros paradigmas como los modelos probabilísticos, ampliamente utilizados para la gestión de incertidumbre [2] y mecanismos por propuestos en el principio de información [170]. Aunque el modelo de representación y manipulación de atributos lingüísticos con semántica adaptable propuesto está desarrollado desde un punto de vista general, su implantación en SGBD actuales está prácticamente limitada a información contextual aportada por el usuario. Dar el salto hacia propuestas de implementación que consideren de forma automática información ambiental aportada por sensores es el siguiente paso en nuestras tareas de desarrollo de software.
- Incluir el histórico de acciones de un usuario como un factor adicional que permita inferir preferencias contextuales tal como se propone en la arquitectura para aplicaciones sensibles al contexto, aprovechando la capacidad de los SGBDOR para gestionar datos temporales. Algunos pasos iniciales en este sentido pueden encontrarse en [102].

Bibliografía

- [1] Abiteboul, Serge, and Richard Hull. "IFO: A formal semantic database model." *ACM Transactions on Database Systems (TODS)* 12, no. 4 (1987): 525-565.
- [2] Aggarwal, C. (Ed.). *Managing and Mining Uncertain Data*. Series: Advances in Database Systems, Vol. 35. Springer, USA, 2009.
- [3] Aguilera, Ana, Livia Borjas, Rosseline Rodriguez, and Leonid Tineo. "Experiences on fuzzy DBMS: Implementation and use." In *Computing Conference (CLEI), 2013 XXXIX Latin American*, pp. 1-8. IEEE, 2013.
- [4] Aguilera, Ana, and Rosseline Rodríguez. "Representación y manipulación de datos médicos en marcha patológica." *Multiciencias* 11, no. 1 (2011).
- [5] Alcala-Fdez, Jesús, and José Alonso. "A Survey of Fuzzy Systems Software: Taxonomy, Current Research Trends and Prospects." *Fuzzy Systems, IEEE Transactions on*. In press, 2015.
- [6] Almazan, Christian Butiu. "Rover: architectural support for exposing and using context." Phd diss. Tesis doctoral presentada ante la Universidad de Maryland. (2010).
- [7] Amini, Amineh, Teh Ying Wah, and Hadi Saboohi. "On density-based data streams clustering algorithms: A survey." *Journal of Computer Science and Technology* 29, no. 1 (2014): 116-141
- [8] Arvanitis, Anastasios, and Georgia Koutrika. "PrefDB: Supporting preferences as first-class citizens in relational databases." *IEEE Transactions on Knowledge and Data Engineering* 26, no. 6 (2014): 1430-1446.
- [9] Atzori, Luigi, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey." *Computer networks* 54, no. 15 (2010): 2787-2805.
- [10] Bahri, Afef, Rafik Bouaziz, and Faiez Gargouri. "Fuzzy ontology implementation and query answering on databases." In *Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American*, pp. 1-6. IEEE, 2009.
- [11] Baldauf, Matthias, Schahram Dustdar, and Florian Rosenberg. "A survey on context-aware systems." *International Journal of Ad Hoc and Ubiquitous Computing* 2, no. 4 (2007): 263-277.
- [12] Barranco González, Carlos David. "Bases de datos objeto-relacionales difusas: modelo, arquitectura y aplicaciones." PhD diss. Tesis doctoral presentada en la Universidad de Granada, 2010.
- [13] Barranco, Carlos D., Jesús R. Campana, and Juan M. Medina. "A real estate management system based on soft computing." In *Applications of Soft Computing*, pp. 31-40. Springer Berlin Heidelberg, 2009.
- [14] Bedny, Gregory, and David Meister. *The Russian theory of activity: Current applications to design and learning*. Psychology Press, 2014.
- [15] Bellotti, Victoria, and Keith Edwards. "Intelligibility and accountability: human considerations in context-aware systems." *Human-Computer Interaction* 16, no. 2-4 (2001): 193-212.
- [16] Berzal, Fernando, Juan C. Cubero, Nicolás Marín, M. Amparo Vila, Janusz Kacprzyk, and S. Zadrozny. "A general framework for computing

- with words in object-oriented programming." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 15, no. supp01 (2007): 111-131.
- [17] Berzal, Fernando, Nicolás Marín, Olga Pons, and M. A. Vila. "Managing fuzziness on conventional object-oriented platforms." *International Journal of Intelligent Systems* 22, no. 7 (2007): 781-803.
- [18] Bolchini, Cristiana, Carlo A. Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. "A data-oriented survey of context models." *ACM Sigmod Record* 36, no. 4 (2007): 19-26.
- [19] Blanco, I. J., N. Marín, O. Pons, and M. A. Vila. "Softening the object-oriented database model: imprecision, uncertainty, and fuzzy types." In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 4, pp. 2323-2328. IEEE, 2001.
- [20] Blanco, Ignacio J., Carmen Martínez-Cruz, and M. Amparo Vila. "Arquitectura para la integración de esquemas relacionales difusos basada en ontologías: una aplicación para la web." In *ESTYLF08: XIV Congreso Español sobre Tecnologías y Lógica Fuzzy*. Langreo-Mieres, España, 2008.
- [21] Blanco, I., C. Martínez-Cruz, and M. A. Vila. "Looking for Information in Fuzzy Relational Databases accessible via the Web." *Handbook of Research on Web Information Systems Quality* (2008): 300-324.
- [22] Blanco, Ignacio J., Carmen Martínez-Cruz, José María Serrano, and M. Amparo Vila. "A first approach to the multipurpose relational database server." *Mathware and Soft Computing* 12, no. 2-3 (2005): 129-153.
- [23] Bolchini, Cristiana, Elisa Quintarelli, and Letizia Tanca. "CARVE: Context-aware automatic view definition over relational databases." *Information Systems* 38, no. 1 (2013): 45-67.
- [24] Bordogna, Gloria, Alberto Leporati, Dario Lucarella, and Gabriella Pasi. "The fuzzy object-oriented database management system." In *Recent issues on fuzzy databases*, pp. 209-236. Physica-Verlag HD, 2000.
- [25] Borjas, Livia, Josué Ramírez, Rosseline Rodríguez, and Leonid Tineo. "Automated System for Tests Preparation and Configuration Using Fuzzy Queries." In *Computational Intelligence*, pp. 199-212. Springer International Publishing, 2015.
- [26] Bosc, Patrick, Bill B. Buckles, Frederick E. Petry, and Olivier Pivert. *Fuzzy databases*. Springer US, 1999.
- [27] Bosc, Patrick, Donald Kraft, and Fred Petry. "Fuzzy sets in database and information systems: Status and opportunities." *Fuzzy Sets and Systems* 156, no. 3 (2005): 418-426.
- [28] Bosc, Patrick, and Olivier Pivert. "SQLf query functionality on top of a regular relational database management system." In *Knowledge Management in Fuzzy Databases*, pp. 171-190. Physica-Verlag HD, 2000.
- [29] Buckles, Billy P., and Frederick E. Petry. "A fuzzy representation of data for relational databases." *Fuzzy sets and systems* 7, no. 3 (1982): 213-226.
- [30] Bunningen van, Arthur H., Ling Feng, and Peter MG Apers. "A context-aware preference model for database querying in an ambient intelligent environment." In *Database and Expert Systems Applications*, pp. 33-43. Springer Berlin Heidelberg, 2006.

-
- [31] Caballero, Amaury, Miriam Alvarez, Jose Luis Abreu, and Eduardo Caballero. "Methodology for classification of municipalities in the state of Hidalgo, Mexico." In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, edited by M. Demiralp et al., No. 10. World Scientific and Engineering Academy and Society, 2008.
- [32] Cadenas, José Tomás, Ana Aguilera, and Leonid Tineo. "Rendimiento de consultas SQLf en arquitecturas débil y fuertemente acopladas." *Multiciencias* 11, no. 4 (2012).
- [33] Cadenas, José Tomás, Nicolás Marín, and M. Amparo Vila. "Fuzzy domains with adaptable semantics in an object-relational DBMS." In *Flexible Query Answering Systems*, pp. 497-508. Springer Berlin Heidelberg, 2011.
- [34] Cadenas, J. T., N. Marín, and M. A. Vila. "Context-Aware Fuzzy Databases." *Applied Soft Computing*, 25 (2014): 215-233.
- [35] Campaña, Jesús R., Juan M. Medina, and Maria A. Vila. "Semantic Data Management Using Fuzzy Relational Databases." In *Flexible Approaches in Data, Information and Knowledge Management*, pp. 115-140. Springer International Publishing, 2014.
- [36] Carrasco, Ramón Alberto, María Amparo Vila, Francisco Araque, Alberto Salguero, and Mari Ángeles Aguilar. "dmFSQL: a Server for Data Mining." In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pp. 888-895. IEEE, 2007.
- [37] Chen, Harry, T. Finin, and A. Joshi. "An intelligent broker architecture for context-aware systems." *PhD proposal in Computer Science, University of Maryland, Baltimore, USA*, 2003.
- [38] Chen, Guanling, and David Kotz. *A survey of context-aware mobile computing research*. Vol. 1, no. 2.1. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
- [39] Codd, Edgar F. "A relational model of data for large shared data banks." *Communications of the ACM* 13, no. 6 (1970): 377-387.
- [40] Cox, Earl. *The fuzzy systems handbook: a practitioner's guide to building, using, maintaining fuzzy systems*. Boston: AP Professional. ISBN 0-12-194270-8, 1994.
- [41] Cuevas, L. "Modelo difuso de bases de datos objeto-relacional: propuesta de implementación en software libre." PhD diss. Tesis Doctoral presentada ante la Universidad de Granada. Granada, España, 2009.
- [42] Cuevas, Luis, Nicolás Marín, Olga Pons, and Maria Amparo Vila. "pg4DB: A fuzzy object-relational system." *Fuzzy Sets and Systems* 159, no. 12 (2008): 1500-1514.
- [43] De Tré, Guy, and Rita De Caluwe. "Level-2 fuzzy sets and their usefulness in object-oriented database modelling." *Fuzzy Sets and Systems* 140, no. 1 (2003): 29-49.
- [44] Delgado, G., V. Aranda, J. Calero, M. Sánchez-Marañón, J. M. Serrano, D. Sánchez, and María Amparo Vila. "Building a fuzzy logic information network and a decision-support system for olive cultivation in Andalusia [Spain]." *Spanish Journal of Agricultural Research* 6, no. 2 (2008): 252-263.

-
- [45] Delgado, Miguel, M. Dolores Ruiz, Daniel Sánchez, and M. Amparo Vila. "On Fuzzy Modus Ponens to Assess Fuzzy Association Rules." In *Enric Trillas: A Passion for Fuzzy Sets*, pp. 269-276. Springer International Publishing, 2015.
- [46] Delgado, Miguel, M. Dolores Ruiz, Daniel Sánchez, and M. Amparo Vila. "Fuzzy quantification: a state of the art." *Fuzzy Sets and Systems* 242 (2014): 1-30.
- [47] Delgado, Miguel, Nicolás Marín, Yilena Pérez, and M. Amparo Vila. "Bipolar queries on fuzzy univalued and multivalued attributes in object databases." *Fuzzy Sets and Systems*, 2015.
- [48] Dey, Anind K. "Understanding and using context." *Personal and ubiquitous computing* 5, no. 1 (2001): 4-7.
- [49] Dey, Anind K., Gregory D. Abowd, and Daniel Salber. "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications." *Human-computer interaction* 16, no. 2 (2001): 97-166.
- [50] Dourish, Paul. "Seeking a foundation for context-aware computing." *Human-Computer Interaction* 16, no. 2-4 (2001): 229-241.
- [51] Dubois, Didier J. *Fuzzy sets and systems: theory and applications*. Vol. 144. Academic press, 1980.
- [52] Dubois, Didier J., Henri Prade, and Ronald R. Yager, eds. *Readings in fuzzy sets for intelligent systems*. Morgan Kaufmann, 2014.
- [53] Galindo, José, ed. *Handbook of research on fuzzy information processing in databases*. IGI Global, 2008.
- [54] Galindo, José, ed. *Fuzzy Databases: Modeling, Design and Implementation: Modeling, Design and Implementation*. IGI Global, 2005.
- [55] Galindo, José, Juan M. Medina, Olga Pons, and Juan C. Cubero. "A server for fuzzy SQL queries." In *Flexible query answering systems*, pp. 164-174. Springer Berlin Heidelberg, 1998.
- [56] Gammoudi, Aymen, Allel Hadjali, and Boutheina Ben Yaghlane. "An Intelligent Flexible Querying Approach for Temporal Databases." In *Intelligent Systems' 2014*, pp. 523-534. Springer International Publishing, 2015.
- [57] Garrido, Carmen, Nicolás Marín, and Olga Pons. "Fuzzy intervals to represent fuzzy valid time in a temporal relational database." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 17, no. supp01 (2009): 173-192.
- [58] George, Roy, Radhakrishnan Srikanth, Frederick E. Petry, and Bill P. Buckles. "Uncertainty management issues in the object-oriented data model." *Fuzzy Systems, IEEE Transactions on* 4, no. 2 (1996): 179-192.
- [59] Golunska, Dominika, Janusz Kacprzyk, and Slawomir Zadrozny. "On efficiency-oriented support of consensus reaching in a group of agents in a fuzzy environment with a cost based preference updating approach." In *Computational Intelligence for Human-like Intelligence (CIHLI), 2014 IEEE Symposium on*, pp. 1-7. IEEE, 2014.
- [60] Goncalves M. & Tineo L. "SQLfi and its Applications". *Avances en Sistemas e Informática*, Vol 5 No. 2. Medellin, Colombia. ISSN 1657-7663, 2008.

-
- [61] Gottwald, Siegfried. *Fuzzy sets and fuzzy logic*. Braunschweig: Vieweg, 1993.
- [62] Hadjali, Allel, Amine Mokhtari, and Olivier Pivert. "A fuzzy-rule-based model for handling contextual preference queries". *International Journal of Computational Intelligence Systems* 5, no. 4 (2012): 775-788.
- [63] Hernández Sampieri, Roberto, Carlos Fernández Collado, and Pilar Baptista Lucio. "Metodología de la investigación." *México: Editorial Mc Graw Hill* (2010).
- [64] Hong, Jason I., and James A. Landay. "An infrastructure approach to context-aware computing." *Human-Computer Interaction* 16, no. 2 (2001): 287-303.
- [65] Hong, Jong-yi, Eui-ho Suh, and Sung-Jin Kim. "Context-aware systems: A literature review and classification." *Expert Systems with Applications* 36, no. 4 (2009): 8509-8522.
- [66] Huynh, Kim Tâm, Béatrice Finance, and Mokrane Bouzeghoub. "Towards an Ambient Data Mediation System." In *IMMoA*, pp. 13-20. 2012.
- [67] Hyman, Steven E. "Diagnosing disorders." *Scientific American* 289, no. 3 (2003): 96-103.
- [68] Ishii, Hiroshi, and Brygg Ullmer. "Tangible bits: towards seamless interfaces between people, bits and atoms." In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pp. 234-241. ACM, 1997.
- [69] Jaime-Castillo, Sergio, Juan M. Medina, Carlos D. Barranco, and Antonio Garrido. "On the use of a fuzzy object-relational database for retrieval of x-rays on the basis of spine curvature pattern similarities." In *Flexible Query Answering Systems*, pp. 436-447. Springer Berlin Heidelberg, 2011.
- [70] Jandoubi, Sabrine, Afef Bahri, Salem Chakhar, and Nadia Yacoubi-Ayadi. "Mapping the Fuzzy Semantic Model into Fuzzy Object Relational database Model." In *Proceedings of the Seventh International Conference on Information, Process, and Knowledge Management, eKNOW 2015*. Lisbon, Portugal. (2015): 138-143.
- [71] Jiang, Yuncheng, Yong Tang, Ju Wang, and Suqin Tang. "Representation and reasoning of context-dependant knowledge in distributed fuzzy ontologies." *Expert Systems with Applications* 37, no. 8 (2010): 6052-6060.
- [72] Jones, Simon, Sukhvinder Hara, and Juan Augusto. "eFRIEND: an ethical framework for intelligent environment development." In *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*, p. 58. ACM, 2014.
- [73] Kaptelinin, Victor, and Bonnie Nardi. "Activity theory in HCI: Fundamentals and Reflections." *Synthesis Lectures Human-Centered Informatics* 5, no. 1 (2012): 1-105.
- [74] Karwowski, Waldemar, and Anil Mital, eds. *Applications of fuzzy set theory in human factors*. Elsevier, 2014.
- [75] Klir, George, and Bo Yuan. *Fuzzy sets and fuzzy logic*. Vol. 4. New Jersey: Prentice Hall, 1995.
- [76] Koyuncu, Murat. "Intelligent fuzzy queries for multimedia databases." *International Journal of Intelligent Systems* 26, no. 10 (2011): 930-951.

-
- [77] Koutrika, Georgia. "Personalized DBMS: an Elephant in Disguise or a Chameleon?" *IEEE Data Eng. Bull.* 34, no. 2 (2011): 27-34.
- [78] Kruse, Rudolf, Joan E. Gebhardt, and F. Klowon. *Foundations of fuzzy systems*. John Wiley & Sons, Inc., 1994.
- [79] Lai, Lien-Fu, Chao-Chin Wu, Yi-Ta Hsieh, and Liang-Tsung Huang. "A Fuzzy Query Approach to Human Resource Web Services." In *e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on*, pp. 461-466. IEEE, 2013.
- [80] Lee, Chang-Shing, Mei-Hui Wang, and Hani Hagraas. "A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation." *Fuzzy Systems, IEEE Transactions on* 18, no. 2 (2010): 374-395.
- [81] Leiva Olivencia, José Luis. "Realidad aumentada bajo tecnología móvil basada en el contexto aplicada a destinos turísticos." PhD diss. Tesis doctoral presentada ante la Universidad de Málaga, España. (2014).
- [82] Levandoski, Justin J., Ahmed Eldawy, Mohamed F. Mokbel, and Mohamed E. Khalefa. "Flexible and extensible preference evaluation in database systems." *ACM Transactions on Database Systems (TODS)* 38, no. 3 (2013): 17.
- [83] Lieberman, Henry, and Ted Selker. "Out of context: Computer systems that adapt to, and learn from, context." *IBM Systems Journal* 39, no. 3.4 (2000): 617-632.
- [84] Liu, Rui, Xiaoli Zhang, and Songpo Li. "Use context to understand user's implicit intentions in Activities of Daily Living." In *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, pp. 1214-1219. IEEE, 2014.
- [85] Ma, Zongmin, ed. *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications: Modeling and Applications*. IGI Global, 2004.
- [86] Ma, Z. M., and Li Yan. "A Literature Overview of Fuzzy Conceptual Data Modeling." *J. Inf. Sci. Eng.* 26, no. 2 (2010): 427-441.
- [87] Ma, Z. M., Wen-Jun Zhang, W. Y. Ma, and G. Q. Chen. "Conceptual design of fuzzy object-oriented databases using extended entity-relationship model." *International Journal of Intelligent Systems* 16, no. 6 (2001): 697-711.
- [88] Ma, Zongmin, Fu Zhang, Li Yan, and Jingwei Cheng. "Querying Fuzzy Description Logics and Ontology Knowledge Bases." In *Fuzzy Knowledge Management for the Semantic Web*, pp. 181-231. Springer Berlin Heidelberg, 2014.
- [89] Magdalena, Luis, José Luis Verdegay, and Francesc Esteva. "Enric Trillas: A Passion for Fuzzy Sets. A Collection of Recent Works on Fuzzy Logic". *Studies in Fuzziness and Soft Computing*. Volume 322, 2015.
- [90] Marín, Nicolás. *Estudio de la vaguedad en los sistemas de bases de datos orientados a objetos: tipos difusos y sus aplicaciones*. Ph diss. Tesis doctoral presentada ante la Universidad de Granada, 2001.
- [91] Marín, N., Pons, O. and Vila, M. A., Fuzzy types: A new concept of type for managing vague structures. *Int. J. Intell. Syst.*, 15: 1061–1085, 2000.
- [92] Marín, N., Pons, O. and Vila, M. A. A strategy for adding fuzzy types to an object-oriented database system. *Int. J. Intell. Syst.*, 16: 863–880, 2001.

-
- [93] Marín, Nicolás, Juan Miguel Medina, Olga Pons, Daniel Sánchez, and M. A. Vila. "Complex object comparison in a fuzzy context." *Information and software Technology* 45, no. 7 (2003): 431-444.
- [94] Marín, Nicolás, and Daniel Sánchez. "On generating linguistic descriptions of time series." *Fuzzy Sets and Systems* (2015).
- [95] Martinenghi, Davide, and Riccardo Torlone. "Querying context-aware databases." In *Flexible Query Answering Systems*, pp. 76-87. Springer Berlin Heidelberg, 2009.
- [96] McNeill, F. Martin, and Ellen Thro. *Fuzzy logic: a practical approach*. Academic Press, 2014.
- [97] Medina, Juan Miguel, Carlos D. Barranco, Jesús R. Campaña, and Sergio Jaime-Castillo. "Generalized fuzzy comparators for complex data in a fuzzy object-relational database management system." In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, pp. 126-136. Springer Berlin Heidelberg, 2010.
- [98] Medina, Juan Miguel, J. Enrique Pons, Carlos D. Barranco, and Olga Pons. "A Fuzzy Temporal Object-Relational Database: Model and Implementation." *International Journal of Intelligent Systems* 29, no. 9 (2014): 836-863.
- [99] Medina, Juan Miguel, Olga Pons, and Maria Amparo Vila. "GEFRED: a generalized model of fuzzy relational databases." *Information sciences* 76, no. 1 (1994): 87-109.
- [100] Medina, Juan Miguel, Sergio Jaime-Castillo, Carlos D. Barranco, and Jesús R. Campana. "On the Use of a Fuzzy Object-Relational Database for Flexible Retrieval of Medical Images." *Fuzzy Systems, IEEE Transactions on* 20, no. 4 (2012): 786-803.
- [101] Miele, Antonio, Elisa Quintarelli, and Letizia Tanca. "A methodology for preference-based personalization of contextual data." In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 287-298. ACM, 2009.
- [102] Miele, Antonio, Elisa Quintarelli, Emanuele Rabosio, and Letizia Tanca. "ADaPT: Automatic Data Personalization based on contextual preferences." In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pp. 1234-1237. IEEE, 2014.
- [103] Miele, Antonio, Elisa Quintarelli, Emanuele Rabosio, and Letizia Tanca. "A data-mining approach to preference-based data ranking founded on contextual information." *Information Systems* 38, no. 4 (2013): 524-544.
- [104] Motro, Amihai. "Accommodating imprecision in database systems: issues and solutions." *ACM SIGMOD Record* 19, no. 4 (1990): 69-74.
- [105] Moreno, María N., Joel P. Lucas, and Vivian F. López. "Fuzzy Data-Mining Hybrid Methods for Recommender Systems." In *Soft Computing for Business Intelligence*, pp. 327-344. Springer Berlin Heidelberg, 2014.
- [106] Myszkowski, Krzysztof. "Multiargument Relationships in Possibilistic Databases." In *Beyond Databases, Architectures and Structures*, pp. 147-156. Springer International Publishing, 2015.
- [107] Nardi, Bonnie A., ed. *Context and consciousness: activity theory and human-computer interaction*. Mit Press, 1996.

-
- [108] Oliver, Nuria M., and Alex P. Pentland. "Towards perceptual intelligence: statistical modeling of human individual and interactive behaviors." PhD diss., Massachusetts Institute of Technology, Dept. of Architecture, 2000.
- [109] Orsi, Giorgio, and Letizia Tanca. "Context modelling and context-aware querying." In *Datalog Reloaded*, pp. 225-244. Springer Berlin Heidelberg, 2011.
- [110] Palella, Santa, and Feliberto Martins. "Metodología de la investigación cuantitativa." *Caracas, Venezuela: Fedupel* (2006).
- [111] Pedrycz, Witold. "Why triangular membership functions?" *Fuzzy sets and Systems* 64, no. 1 (1994): 21-30.
- [112] Perera, Charith, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. "Context aware computing for the internet of things: A survey." *Communications Surveys & Tutorials, IEEE* 16, no. 1 (2014): 414-454.
- [113] Petry, Frederick E. "Fuzzy Set Theory Utility for Database and Information Systems." In *On Fuzziness*, pp. 547-552. Springer Berlin Heidelberg, 2013.
- [114] Petry, Frederick E. *Fuzzy Databases: Principles and Applications*. Kluwer Academic Publishers, 1997.
- [115] Petry, Frederick E., and Ronald R. Yager. "Fuzzy Concept Hierarchies and Evidence Resolution." *Fuzzy Systems, IEEE Transactions on* 22, no. 5 (2014): 1151-1161.
- [116] Pitoura, Evaggelio, Kostas Stefanidis, and A. B. Zaslavsky. "Context in Databases." Technical Report TR 2004-19, Department of Computer Science, *University of Ioannina, Greece* (2004): 1-19.
- [117] Pivert, Olivier, and Grégory Smits. "Plethoric answers to fuzzy queries: A reduction method based on query mining." In *Foundations of Intelligent Systems*, pp. 295-304. Springer International Publishing, 2014.
- [118] Pons, Jose Enrique, Nicolás Marín, Olga Pons, Christophe Billiet, and Guy De Tré. "A relational model for the possibilistic valid-time approach". *International Journal of Computational Intelligence Systems* 5, no. 6 (2012): 1068-1088.
- [119] Prade, Henri, and Claudette Testemale. "The possibilistic approach to the handling of imprecision in database systems." *IEEE Data Eng. Bull.* 12, no. 2 (1989): 4-10.
- [120] Quintarelli, Elisa, and Emanuele Rabosio. "Discovering Contextual Association Rules in Relational Databases." In *New Trends in Databases and Information Systems*, pp. 193-202. Springer International Publishing, 2014.
- [121] Quintarelli, Elisa, Emanuele Rabosio, and Letizia Tanca. "A principled approach to context schema evolution in a data management perspective." *Information Systems* 49 (2015): 65-101.
- [122] Raju, K. V. S. V. N., and Arun K. Majumdar. "Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems." *ACM Transactions on Database Systems (TODS)* 13, no. 2 (1988): 129-166.

-
- [123] Rodrigues, R. D., A. J. O. Cruz, and R. T. Cavalcante. "Aliança: A proposal for a fuzzy database architecture incorporating XML." *Fuzzy sets and systems* 160, no. 2 (2009): 269-279.
- [124] Rooks, Patrick, Markus Endres, Stefan Mandl, and Werner Kießling. "Composition and efficient evaluation of context-aware preference queries." In *Database Systems for Advanced Applications*, pp. 81-95. Springer Berlin Heidelberg, 2012.
- [125] Rossazza, Jean-Paul, Didier Dubois, and Henri Prade. "A hierarchical model of fuzzy classes." *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models, World Scientific* (1997): 21-61.
- [126] Roussos, Ioannis N., and Timos Sellis. "A model for context aware relational databases." Technical Report. National Technical University of Athens (2008).
- [127] Ruspini, E. "Imprecision and uncertainty in the entity-relationship model." *Fuzzy Logic and Knowledge Engineering* (1986): 3-17.
- [128] Sadri, Fariba. "Ambient intelligence: A survey". *ACM Computing Surveys (CSUR)* 43, no. 4 (2011): 36.
- [129] Salguero, Alberto, Francisco Araque, Ramón Alberto Carrasco, M. A. Vila, and Luis Martínez. "Applying fuzzy data mining for soaring area selection." In *Computational and Ambient Intelligence*, pp. 597-605. Springer Berlin Heidelberg, 2007.
- [130] Sarkar, S. and Dey, D. Relational Models and Algebra for Uncertain Data. In Aggarwal, C. (Ed.): *Managing and mining Uncertain Data*. Series: Advances in Database Systems, Vol. 35. Springer, USA, 2009.
- [131] Schmidt, Albrecht. "Implicit human computer interaction through context". *Personal technologies* 4, no. 2-3 (2000): 191-199.
- [132] Schmidt, Andreas. "A layered model for user context management with controlled aging and imperfection handling." In *Modeling and Retrieval of Context*, pp. 86-100. Springer Berlin Heidelberg, 2006.
- [133] Schneider, Oliver S., Karon E. MacLean, Kerem Altun, Idin Karuei, and Michael Wu. "Real-time gait classification for persuasive smartphone apps: structuring the literature and pushing the limits." In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 161-172. ACM, 2013.
- [134] Seising, Rudolf. *The Fuzzification of Systems: The Genesis of Fuzzy Set Theory and its Initial Applications-Developments up to the 1970s*. Springer Publishing Company, Incorporated, 2007.
- [135] Sheno, Sujeet, Austin Melton, and L. T. Fan. "An equivalence classes model of fuzzy relational databases." *Fuzzy Sets and Systems*, 38, no. 2 (1990): 153-170.
- [136] Sriramdas, V., Sanjay Kumar Chaturvedi, and Heeralal Gargama. "Fuzzy arithmetic based reliability allocation approach during early design and development." *Expert Systems with Applications* 41, no. 7 (2014): 3444-3449.
- [137] Stefanidis, Kostas, Georgia Koutrika, and Evaggelia Pitoura. "A survey on representation, composition and application of preferences in database systems." *ACM Transactions on Database Systems (TODS)* 36, no. 3 (2011): 19.

-
- [138] Strang, Thomas, and Claudia Linnhoff-Popien. "A context modeling survey." In *Workshop Proceedings*. 2004.
- [139] Suchman, Lucy A. *Plans and situated actions: the problem of human-machine communication*. Cambridge university press, 1987.
- [140] Tahani, Valiollah. "A conceptual framework for fuzzy query processing—a step toward very intelligent database systems." *Information Processing & Management* 13, no. 5 (1977): 289-303.
- [141] Tanca, Letizia, Cristiana Bolchini, Elisa Quintarelli, Fabio A. Schreiber, and Giorgio Orsi. "Problems and opportunities in context-based personalization." *Proceedings of the VLDB Endowment* 4, no. 11 (2011): 1-4.
- [142] Terano, Toshiro, Kiyoji Asai, and Michio Sugeno, eds. *Applied fuzzy systems*. Academic Press, 2014.
- [143] Testemale, Claudia. "Fuzzy relational databases—a key to expert systems." (1986): 272-273.
- [144] Tho, Quan Thanh, Siu Cheung Hui, Alvis Cheuk M. Fong, and Tru Hoang Cao. "Automatic fuzzy ontology generation for semantic web." *Knowledge and Data Engineering, IEEE Transactions on* 18, no. 6 (2006): 842-856.
- [145] Touzi, Amel Grissa, and Habib Ounalli. "New approach for flexible query using the knowledge discovered in large database." *International Journal of Computational Systems Engineering* (2015).
- [146] Touzi, Amel Grissa, and Mohamed Ali Ben Hassine. "New architecture of fuzzy database management systems." *The International Arab Journal of Information Technology* 6, no. 2 (2009).
- [147] Umamo, Motohide. "FREEDOM-0: a fuzzy database system." *Fuzzy information and decision processes* (1982): 339-347.
- [148] Urrutia, Angélica, José Galindo, and Alejandro Sepúlveda. "Implementación de una base de datos difusa con First-2 y PostgreSQL." In *XV Congreso Español sobre Tecnologías y Lógica Fuzzy ESTYLF*, pp. 199-204. 2010.
- [149] Urrutia, Angélica, and José Galindo. "Fuzzy Database Modeling: An Overview and New Definitions." *Soft Computing Applications for Database Technologies: Techniques and Issues* (2010): 1.
- [150] Van Gyseghem, Nancy, and Rita De Caluwe. "The UFO Database Model: Dealing with Imperfect Information." *Fuzzy and uncertain object-oriented databases: Concepts and models* 13 (1997): 123-185.
- [151] Van Hung Le, Fei Liu, and Hongen Lu. "A Data Model for Fuzzy Linguistic Databases with Flexible Querying." *AI 2009: Advances in Artificial Intelligence*: 495.
- [152] Vandenberghe, R. M., and R. M. De Caluwe. "An entity-relationship approach to the modelling of vagueness in databases." In *Symbolic and Quantitative Approaches to Uncertainty*, pp. 338-343. Springer Berlin Heidelberg, 1991.
- [153] Vila, M. A., J. C. Cubero, J. M. Medina, and O. Pons. "A logic approach to fuzzy relational databases." *International Journal of Intelligent Systems* 9, no. 5 (1994): 449-460.
- [154] Vila, M. A., J. C. Cubero, J. M. Medina, and O. Pons. "A conceptual approach for dealing with imprecision and uncertainty in object-based data

- models." *International Journal of Intelligent Systems* 11, no. 10 (1996): 791-806.
- [155] Vila, M. A., J. C. Cubero, J. M. Medina, and O. Pons. *A fuzzy object-oriented data model represented by means of a semantic data model*. Tech. Rep. DECSAI-98-00-00. Department of Computer Science and Artificial Intelligence-University of Granada, 1998.
- [156] Weiser, Mark. "The computer for the 21st century." *Scientific American* 265, no. 3 (1991): 94-104.
- [157] Wenzel, Florian, Martin Soutschek, and Werner Kießling. *A Preference SQL Approach to Improve Context-Adaptive Location-Based Services for Outdoor Activities*. Springer Berlin Heidelberg, 2012.
- [158] Yazici, Adnan, and Roy George. *Fuzzy database modeling*. Vol. 26. Springer Science & Business Media, 1999.
- [159] Yan, Li, and Z. M. Ma. "Modeling fuzzy information in fuzzy extended entity-relationship model and fuzzy relational databases." *Journal of Intelligent and Fuzzy Systems* 27, no. 4 (2014): 1881-1896.
- [160] Zadeh, Lotfi A. "Fuzzy sets." *Information and control* 8, no. 3 (1965): 338-353.
- [161] Zadeh, Lotfi A. "The concept of a linguistic variable and its application to approximate reasoning—I." *Information sciences* 8, no. 3 (1975): 199-244.
- [162] Zadeh, Lotfi A. "The concept of a linguistic variable and its application to approximate reasoning—II." *Information sciences* 8, no. 4 (1975): 301-357.
- [163] Zadeh, Lotfi A. "The concept of a linguistic variable and its application to approximate reasoning-III." *Information sciences* 9, no. 1 (1975): 43-80.
- [164] Zadeh, Lotfi A. "Soft computing and fuzzy logic." *IEEE software* 11, no. 6 (1994): 48-56.
- [165] Zadeh, Lotfi A. "Fuzzy logic= computing with words." *Fuzzy Systems, IEEE Transactions on* 4, no. 2 (1996): 103-111.
- [166] Zadeh, Lotfi Asker, Bo Yuan, and George J. Klir. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh*. World Scientific Publishing Co., Inc., 1996.
- [167] Zadeh, Lotfi A. "Fuzzy sets as a basis for a theory of possibility." *Fuzzy sets and systems* 100 (1999): 9-34.
- [168] Zadeh, Lotfi A. "A New Direction in AI Toward a Computational Theory of Perceptions." In *Technologies for Constructing Intelligent Systems 1*, pp. 3-20. Physica-Verlag HD, 2002.
- [169] Zadeh, Lotfi A. "Toward a restriction-centered theory of truth and meaning (RCT)." *Information Sciences* 248 (2013): 1-14.
- [170] Zadeh, Lotfi A. "The information principle." *Information Sciences* 294 (2015): 540-549.
- [171] Zadeh, Lotfi Asker, Bo Yuan, and George J. Klir. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh*. World Scientific Publishing Co., Inc., 1996.
- [172] Zadeh, Lotfi A., King-Sun Fu, and Kokichi Tanaka, eds. *Fuzzy Sets and Their Applications to Cognitive and Decision Processes: Proceedings of the US-Japan Seminar on Fuzzy Sets and Their Applications, Held at the*

-
- University of California, Berkeley, California, July 1-4, 1974.* Academic Press, 2014.
- [173] Zadrozny, Sławomir, Janusz Kacprzyk, Mateusz Dzielic, and Guy De Tré. "Contextual bipolar queries." In *Advance Trends in Soft Computing*, pp. 421-428. Springer International Publishing, 2014.
- [174] Zemankova-Leech, Maria, and Abraham Kandel. *Fuzzy relational data bases: a key to expert systems*. Vol. 84. Verl. TÜV Rheinland, 1984.
- [175] Zhao, Faxin, Zong-Min Ma, and Li Yan. "Fuzzy clustering based on vague relations." In *Fuzzy Systems and Knowledge Discovery*, pp. 79-88. Springer Berlin Heidelberg, 2006.
- [176] Zimmermann, Andreas, Andreas Lorenz, and Reinhard Oppermann. "An operational definition of context." In *Modeling and using context*, pp. 558-571. Springer Berlin Heidelberg, 2007.
- [177] Zvieli, Arie, and Peter P. Chen. "Entity-relationship modeling and fuzzy databases." In *Proceedings of the Second International Conference on Data Engineering*, pp. 320-327. IEEE Computer Society, 1986.

Apendice A

Formularios Examen Físico Articular

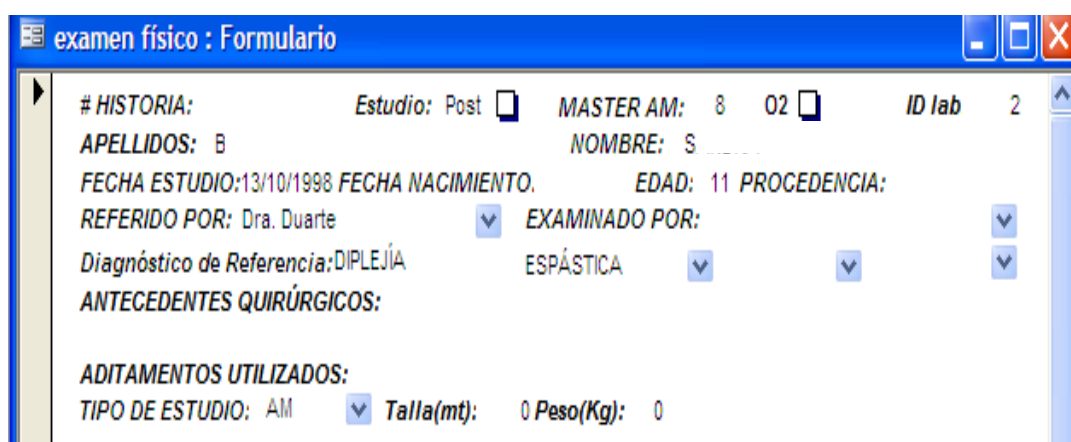
A continuación se describen los Formularios utilizados actualmente para registrar datos de los pacientes y del Examen Físico que se realiza en el Laboratorio de Marcha en el Hospital Ortopédico Infantil de Venezuela.

Se presentan:

I.- Datos de Identificación del paciente

II.- Examen Físico Articular.

I. Datos de Identificación:



The screenshot shows a software window titled "examen físico : Formulario". The form contains the following fields and options:

- # HISTORIA: [input field]
- Estudio: Post
- MASTER AM: 8 02
- ID lab: 2
- APELLIDOS: B
- NOMBRE: S
- FECHA ESTUDIO: 13/10/1998
- FECHA NACIMIENTO: [input field]
- EDAD: 11
- PROCEDENCIA: [input field]
- REFERIDO POR: Dra. Duarte
- EXAMINADO POR: [dropdown menu]
- Diagnóstico de Referencia: DIPLEJÍA
- ESPÁSTICA [dropdown menu]
- ANTECEDENTES QUIRÚRGICOS: [dropdown menu]
- ADITAMENTOS UTILIZADOS: [input field]
- TIPO DE ESTUDIO: AM [dropdown menu]
- Talla(mt): [input field]
- Peso(Kg): [input field]

1. **#HISTORIA:** Número de la historia del paciente en el **Archivo de Historias Médicas** del H.O.I. La selección sobre el formulario se reinicia ubicándose en el campo #HISTORIA.
2. **ESTUDIO:** Presenta dos opciones: PRE y POST
 - 2.1 PRE significa Preoperatorio.
 - 2.2 POST significa Postoperatorio.
3. **MASTER AM:** Se refiere al **número** del cassette de VHS empleado para grabar el video bidimensional, con vistas sagital y frontal durante el análisis de la marcha.
4. **ID LAB:** Número de historia en el laboratorio de marcha.

5. **APELLIDOS:** Apellido del paciente.
6. **NOMBRES:** Nombre del paciente.
7. **FECHA ESTUDIO:** La fecha en la que se realizó el examen físico (día/mes/año).
8. **FECHA NACIMIENTO:** La fecha de nacimiento del paciente (día/mes/año).
9. **EDAD:** Edad del paciente (en años).
10. **PROCEDENCIA:** Lugar donde vive el paciente.
11. **REFERIDO POR:** Médico que envió al paciente al Laboratorio de Marcha. Se presenta una lista con las siguientes opciones:
 - Apellidos de los médicos que han referido pacientes previamente. (Por ejemplo: Dr. Posadas, Dr. Rupcich,...)
 - N/A: No Aplica.
 - Privado

The screenshot shows a software window titled "examen físico : Formulario". The form contains the following fields and values:

- # HISTORIA: Estudio: Post MASTER AM: 8 ID lab 2
- APELLIDOS: BRIC... NOMBRE: SANL...
- FECHA ESTUDIO: 13/10/1998 FECHA NACIMIENTO: EDAD: 11 PROCEDENCIA:
- REFERIDO POR: **Dra. Duarte** (dropdown menu is open showing: Dra. Fernández, Dr. Prado, Dr. Rodríguez, Hospital Clínico Universitario, Privado, N/A, Dr. Garcés)
- EXAMINADO POR: (dropdown menu)
- Diagnóstico de Referencia: ESPÁSTICA (dropdown menu)
- ANTECEDENTES QUIRÚRGICOS: (dropdown menu)
- ADITAMENTOS UTILIZADOS: Hospital Clínico Universitario
- TIPO DE ESTUDIO: Privado
- 0 Peso(Kg): 0
- Selectividad/Fuerza Muscular: Tono muscular

12. **EXAMINADO POR:** Fisioterapeuta que realiza el Examen Físico al paciente. Son cinco fisioterapeutas diferentes, uno en cada ítem.

The screenshot shows the same software window "examen físico : Formulario". The form contains the following fields and values:

- # HISTORIA: Estudio: Post MASTER AM: 8 ID lab 2
- APELLIDOS: NOMBRE:
- FECHA ESTUDIO: 13/10/1998 FECHA NACIMIENTO: EDAD: 11 PROCEDENCIA:
- REFERIDO POR: Dra. Duarte (dropdown menu)
- EXAMINADO POR: **DANIROA URBANO** (dropdown menu is open showing: DANIROA URBANO, MARIANA BARRETO, RICARDO GONZÁLEZ, LISET ALVAREZ, LIZA BARRIOS)
- Diagnóstico de Referencia: DIPLEJÍA
- ANTECEDENTES QUIRÚRGICOS: (dropdown menu)
- ADITAMENTOS UTILIZADOS:
- TIPO DE ESTUDIO: AM Talla(mt): 0 Peso(Kg): 0

13. **DIAGNÓSTICO DE REFERENCIA:** Diagnóstico con el que llega al paciente al laboratorio de marcha. El ítem (EN ESTUDIO) significa que no se ha determinado la patología que presenta el paciente.

examen físico : Formulario

HISTORIA: Estudio: Post MASTER AM: 8 O2 ID lab 2

APELLIDOS: BI NOMBRE: S,

FECHA ESTUDIO:13/10/1998 FECHA NACIMIENTO. EDAD: 11 ICIA:

REFERIDO POR: Dra. Duarte EXAMINADO POR:

Diagnóstico de Referencia: PERTEJA

ANTECEDENTES QUIRÚRGICOS: PERTHES
PIE EQUINO VARO
SÍNDROME DE STRUGER WEBER
TORSIÓN TIBIAL INTERNA
EN ESTUDIO
MIELODISPLASIA
SÍNDROME DE DOWN
MIOPATÍA

ADITAMENTOS UTILIZADO

TIPO DE ESTUDIO: AM

CADERA

Flexión

14. **ESPÁSTICA:** Se refiere a la presencia de rigidez en el tono muscular del paciente. Se presentan como opciones los lados que tiene comprometidos el paciente.

examen físico : Formulario

HISTORIA: Estudio: Post MASTER AM: 8 O2 ID lab 2

APELLIDOS: BRICEÑ NOMBRE: SANDF

FECHA ESTUDIO:13/10/1998 FECHA NACIMIENTO. EDAD: 11 PROCEDENCIA:

REFERIDO POR: Dra. Duarte EXAMINADO POR:

Diagnóstico de Referencia: DIPLEJÍA ESPÁSTICA

ANTECEDENTES QUIRÚRGICOS: DERECHO
IZQUIERDO
BILATERAL

ADITAMENTOS UTILIZADOS:

TIPO DE ESTUDIO: AM Talla(mt): 0 Peso(Kg): 0

15. **ANTECEDENTES QUIRÚRGICOS:** Se refiere a las intervenciones quirúrgicas a las que se ha sometido el paciente antes de llegar al laboratorio de marcha. Se incluyen los pacientes operados, luego del primer análisis de marcha para hacer el estudio Postoperatorio.
16. **ADITAMENTOS UTILIZADOS:** Son los dispositivos que utiliza el paciente para mejorar la marcha indicados por los médicos referentes. Por ejemplo: andaderas, bastón, muletas, férulas, AFO, etc.
17. **TIPO DE ESTUDIO:** Lista con el tipo de estudio a realizar:
AM (análisis de marcha) , C (cinética), C+K (cinética y consumo de energía), C+ EMGD (cinética y electromiografía dinámica), EMGD PL, EMGD (Electromiografía Dinámica), K (consumo de energía), O2 (consumo de oxígeno), VB (video bidimensional).

examen físico : Formulario

HISTORIA: Estudio: Post MASTER AM: 8 02 ID lab 2
 APELLIDOS: BRICEÑ NOMBRE: SANDF
 FECHA ESTUDIO: 13/10/1998 FECHA NACIMIENTO: EDAD: 11 PROCEDENCIA:
 REFERIDO POR: Dra. Duarte EXAMINADO POR:
 Diagnóstico de Referencia: DIPLEJÍA ESPÁSTICA ZUERDO
 ANTECEDENTES QUIRÚRGICOS:

ADITAMENTOS UTILIZADOS:
 TIPO DE ESTUDIO: AM Talla(mt): 0 Peso(Kg): 0

CADERA	Rango Articular			Selectividad/Fuerza Muscular		Tono muscular	
	Izq	Der	Normal	Izq	Der	Izq	Der
Flexión	110	110	(120°-130°)	4/3	4/3	0	0
Extensión	DLN	DLN	(20°-30°)	3/2	3/3	0	0
Abducción(0° flex)	60	60	(45°-50°)	0/2	0/2	0	0
Abducción(90° flex)	45	45	(70°-80°)				
Adducción			(20°-30°)	2/4	2/3	0	0

18. **Talla:** Altura del paciente medida en metros.

19. **Peso:** Peso del paciente medido en kilogramos.

II. Examen Físico Articular: Se evalúan en las articulaciones (cadera, rodilla, tobillo y pie) las características biomecánicas de los miembros inferiores de los pacientes.

examen físico : Formulario

CADERA	Rango Articular			Selectividad/Fuerza Muscular		Tono muscular	
	Izq	Der	Normal	Izq	Der	Izq	Der
Flexión	115	110	(120°-130°)	4/3	4/3	0	0
Extensión	DLN	DLN	(20°-30°)	3/2	3/3	0	0
Abducción(0° flex)	65	60	(45°-50°)	0/2	0/2	0	0
Abducción(90° flex)	45	45	(70°-80°)				
Adducción			(20°-30°)	2/4	2/3	0	0

	Izq	Der		Izq	Der
Anteversión Femoral	35	40	(20°-25°)	Eje bimalleolar	16 12 (15°-25°)
Rotación interna	50	48	(30°-35°)	Angulo Pie Muslo	10 12 (15°)
Rotación externa	37	43	(40°-45°)		(10°)

RODILLA	Izq	Der		Izq	Der	Izq	Der
Flexión	DLN	DLN	(120°-130°)	4/4	4/4	0	0
Extensión	180	-12	(0°)	3/4	3/4	0	0
Ang. Poplíteo	-51	-40	(0°)				
" " Thomas cont.			(0°)				
Flex cad-rodilla 0°	-38	-51	(60°-70°)				

TOBILLO	Izq	Der		Izq	Der	Izq	Der
Dorsiflexión (90°)	+15	+15	(20°-25°)	3/4	3/4	0	0
Dorsiflexión (0°)	+20	+10	(15°-20°)				
Flexión Plantar	120	125	(45°-50°)	3/3	3/3	0	0
Inversión	0	0		0/1	0/1	0	0
Eversión	0	0		1/0	1/0	0	0

PIE	SIN APOYO		CON APOYO	
	Izq	Der	Izq	Der
Antepié Reducible	<input type="checkbox"/>	<input type="checkbox"/>		
Quiebre medio pie Reducible	<input type="checkbox"/>	<input type="checkbox"/>		
Retropié	<input type="checkbox"/>	<input type="checkbox"/>		

MEDIDAS BIOMÉTRICAS		
DISTANCIA EIAS:		0
LONGITUD MS IS	Izq	Der
	0	0
DIÁMETRO BICONDILEO		
DIÁMETRO BIMALLEOLAR		

En cada articulación se evalúa:

- El movimiento característico asociado a ella. Por ejemplo, la cadera es una articulación denominada enartrosis debido a que realiza todos los movimientos articulares, tales como: flexión, extensión, abducción, aducción, rotación interna, rotación externa. Para ello se efectúan pruebas especiales que se describen más adelante.
- El rango articular, selectividad/fuerza muscular y tono muscular en ambos miembros.
 1. **Rango Articular:** Límites de grados en que el paciente logra realizar el movimiento con cualquier articulación.
 2. **Selectividad/Fuerza Muscular:** Selectividad es la capacidad de realizar el movimiento medida en una escala del 0 al 5 en un músculo. La fuerza muscular es la capacidad de un músculo en una escala de 0 al 5 de realizar un movimiento venciendo o no la gravedad y una fuerza que se oponga a dicho movimiento. Es una graduación cualitativa.
 3. **Tono Muscular:** Capacidad de contracción de un músculo, se mide en escala de 1 a 4, siendo (1) atonía (falta de tono) y (4) hipertonía (exceso de tono) con espasticidad.
 4. **Normal:** Los límites de grados normales en que se mueve dicha articulación cuando hace el movimiento, ejemplo: flexión de cadera (normal de 120 a 130 grados).
 5. **Izq. y der. :** Miembro inferior izquierdo y miembro inferior derecho.

A continuación se describen las pruebas especiales que se realizan en cada una de las articulaciones:

Cadera: Articulación que une el fémur con la pelvis.

Flexión de cadera: Cuando el miembro inferior se acerca al tronco anterior en el plano longitudinal.

Extensión de cadera: Cuando el miembro inferior se acerca al tronco posterior en el plano longitudinal.

Abducción 0 grados flex: Cuando el miembro inferior se aleja del tronco en el plano sagital con la rodilla en 0 grados de flexión.

Abducción 90 grados flex: Cuando el miembro inferior se aleja del tronco en el plano sagital con la rodilla en 90 grados de flexión.

Aducción: Cuando el miembro se acerca al tronco en el plano sagital.

Anteversión Femoral: El ángulo que forma el cuello del fémur con la diafisis femoral en el plano transversal, lo normal es de 20 a 25 grados. Si esta disminuido se llama retroversión femoral y el paciente camina con los pies hacia fuera y si esta aumentado se llama anteversión femoral y el paciente camina con los pies hacia adentro.

Rotación Interna de cadera: Cuando el miembro inferior gira hacia adentro con respecto al plano transversal.

Rotación Externa de cadera: Cuando el miembro inferior gira hacia fuera con respecto al plano transversal.

Eje Bimaleolar: El eje que forma los maleolos tibial y peroneal con respecto al eje del fémur.

Angulo Pie Muslo: El ángulo que forma el eje del pie con el eje del muslo.

Rodilla: Articulación que une el fémur con la Tibia.

Flexión de rodilla: Cuando la tibia se acerca al fémur.

Extensión de rodilla: Cuando la tibia se aleja del fémur.

Angulo Poplíteo: El ángulo que forma el eje del fémur con la tibia.

Thomas cont.: El ángulo poplíteo medido con la prueba de Thomas continuada.

Flexión cad-rod 0 grados: Flexión de la cadera con la rodilla a 0 grados de flexoextensión.

Tobillo: Articulación que une la Tibia y el Peroné con el Pie.

Dorsiflexión 90 grados: Movimiento del pie acercándose al plano longitudinal del tobillo con la rodilla a 90 grados de flexión.

Dorsiflexión 0 grados: Movimiento del pie acercándose al plano longitudinal del tobillo con la rodilla a 0 grados de flexo extensión.

Flexión Plantar: Movimiento del pie alejándose del plano longitudinal del tobillo.

Inversión: Cuando el pie se acerca al plano sagital del tobillo.

Eversión: Cuando el pie se aleja del plano sagital del tobillo.

Pie: Parte del cuerpo que permite al individuo el apoyo en el piso. Se evalúa la forma del pie con apoyo o sin apoyo en el suelo, esto con el fin de saber si es un pie plano valgo, varo, equino, etc.

Antepié: La parte del pie formada por los dedos y los metatarsianos.

Medio pie: La parte del pie formada por los huesos cuboides y escafoides.

Retropié: La parte del pie formada por el astrágalo y el calcáneo.

Reductible: Si es capaz de normalizar con el movimiento contrario a la deformidad el pie deforme, por ejemplo pie equino reductible o no reductible.

Quiebre mediopié: Si el mediopié es capaz de doblarse hacia adentro, por ejemplo en el pie flácido.

Medidas Biométricas: sirve para medir si hay disimetría de miembros inferiores, rotación de la pelvis, etc.

Distancia EIAS: Distancia entre las dos espinas iliacas anterosuperiores; es decir, longitud de la pelvis.

Longitud MS Is: Longitud de miembros inferiores.

Diámetro Bicondileo: Diámetro entre los dos cóndilos femorales ((prominencias del hueso): interno y externo.

Diámetro Bimaleolar: Diámetro entre los maléolos tibial y perineal (huesos del tobillo que se encuentran a los lados del pie).

Pruebas Especiales y Reflejos:

<u>REFLEJOS</u>		<u>CLONUS</u>		<u>PRUEBAS ESPECIALES</u>		<u>Izq</u>	<u>Der</u>
<u>OSTEOTENDINOSOS</u>				THOMAS (PSOAS)	--	--	
DER: HIPER	▼	DER: AUSENTE	▼	ADDUCTORES	--	--	
IZQ: HIPER	▼	IZQ: AUSENTE	▼	PHELP	--	--	
				ELYS-DUNCAN	--	--	
				SILVERSKIOLD	--	--	
				CONFUSIÓN	--	--	
<u>OBSERVACIONES:</u>							

Registro: 1 de 1303

Reflejos osteotendinosos: Movimiento involuntario que se produce al golpear con un martillo de reflejos el tendón de un músculo, si es positivo nos indica que raíz nerviosa produce el reflejo. Puede ser medido de 0 a 4, siendo 0 ausente, 1 hiporefléxico, 2 normal, 3 hiperrefléxico y 4 hiperrefléxico con clonus.

Clonus: es el movimiento tipo fasciculación que se produce cuando un reflejo hipertónico aparece; es decir, son los movimientos seguidos que se producen al ocasionar un reflejo.

Pruebas Especiales: Son pruebas realizadas en el examen físico que nos permiten discriminar que músculo se encuentra espástico, según la prueba realizada.

Prueba de Thomas (Psoas): Cuando hay contractura en flexión de la cadera por espasticidad, sucede que al hacer flexión de cadera, con un paciente acostado en decúbito supino, se lleva la rodilla hasta el pecho para aplanar el raquis lumbar, el paciente la realiza y la pierna recta contralateral, en vez de permanecer en la mesa de exploración, se levanta.

Prueba de Aductores: Se le pide al paciente que se sienta y adopte la posición de “mariposa” es decir con las plantas de los pies colocadas una frente a la otra, nos permite ver si hay asimetrías de una pierna con respecto a la otra, es positivo si la hay e indica poca flexibilidad de los aductores. Otro método es colocar al paciente en decúbito supino. Si hay contractura la pierna afectada formará un ángulo menor de 90 grados con una línea que une las dos espinas iliacas anterosuperiores; el médico intenta equilibrar la pelvis y esta última se moverá hacia arriba en el lado afectado.

Prueba de Phelps: Es para verificar si hay acortamiento del recto interno. Se coloca al paciente en decúbito prono, se flexiona la rodilla y se hace abducción del

miembro y extensión de la rodilla, la prueba es positiva si la pierna se va en aducción.

Prueba de ElyDuncan: El paciente se acuesta en decúbito prono, y el médico flexiona la rodilla, el paciente flexiona la cadera del mismo lado, lo que indica que el músculo recto anterior esta tenso de ese lado y la prueba es positiva.

Prueba de Silverskiolds: Cuando hay contractura en flexión plantar se dice que la prueba es positiva cuando existe la contractura en flexión plantar cuando la rodilla esta en extensión máxima

Prueba de Confusión: Se observa si hay sinergia flexora con compromiso neurológico central. Se sienta el paciente y se le pide que flexione la cadera, el paciente hace la flexión de cadera con el pie en dorsiflexión y la prueba es positiva cuando no puede hacer la flexión de cadera con el pie en flexión plantar.

Observaciones: Es para colocar si el paciente presenta otra patología además de la alteración en la marcha. Otras observaciones que quiera el examinador colocar en el examen físico.

Apéndice B

Diccionario de Datos del Examen Físico Articular del Hospital Ortopédico Infantil

User_tab

Atributo	Tipo de Dato Longitud	PK	FK
NickName	UserDef_objtyp	x	

Usuarios

Atributo	Tipo de Dato Longitud	PK	FK
NickName	UserDef_objtyp	x	

Domain_tab

Atributo	Tipo de Dato Longitud	PK	FK
dom_name	Fuzzy_objtyp	x	

Context_tab

Atributo	Tipo de Dato Longitud	PK	FK
Label	Varchar2 50	x	
User_name	Varchar2 50	x	
Dom_name	Varchar2 50	x	Referencia a Domain_tab
Trapezoid	Trapezoid_objtyp		

Medico

Atributo	Tipo de Dato Longitud	PK	FK
usuario	Medico_t	x	

Paciente

Atributo	Tipo de Dato Longitud	PK	FK
CI	Paciente_t	x	
ID Historial	Paciente_t	x	

D_Tono_Muscular

Atributo	Tipo de Dato Longitud	PK	FK
valor	dominio_fijo_t	x	

etiqueta_tono_muscular

Atributo	Tipo de Dato Longitud	PK	FK
etiq	etiqueta_t	x	

semejanza_fijo_etiqueta

Atributo	Tipo de Dato Longitud	PK	FK
etiqueta	semejanza_fijo_etiqueta_t	x	Referencia a etiqueta_Tono_Muscular(etiq)
dominio	semejanza_fijo_etiqueta_t	x	Referencia a D_Tono_Muscular(valor)
usuario	semejanza_fijo_etiqueta_t	x	Referencia a Usuarios(NickName)
dom_name	semejanza_fijo_etiqueta_t	x	Referencia a Domain_tab(dom_name)

etiqueta_carac_marcha

Atributo	Tipo de Dato Longitud	PK	FK
etiq	etiqueta_t	x	

semejanza_etiquetas

Atributo	Tipo de Dato Longitud	PK	FK
etiqueta_1	semejanza_etiquetas_t	x	Referencia a etiqueta_carac_marcha(etiq)
etiqueta_2	semejanza_etiquetas_t	x	Referencia a etiqueta_carac_marcha(etiq)
usuario	semejanza_etiquetas_t	x	Referencia a Domain_tab(dom_name)
dom_name	semejanza_etiquetas_t	x	Referencia a Usuarios(NickName)

dispositivo

Atributo	Tipo de Dato Longitud	PK	FK
etiq	etiqueta_t	x	

dispositivos_usados

Atributo	Tipo de Dato Longitud	PK	FK
paciente	dispositivos_usados_t	x	Referencia a Paciente(CI,ID_Historial)
dispositivo	dispositivos_usados_t	x	Referencia a dispositivo(etiq)

Atributo	Tipo de Dato Longitud	PK	FK
ID Historial	Number 6		Paciente(ID_Historial)
Estudio PrPs	Varchar 2		
ID Lab	Number 6		
Fecha Examen	Date	x	
Edad	Number 2		
Medico Int	Varchar 50		Medico(usuario)
Medico Fisio	Varchar 50		Medico(usuario)
Diagn_ref	Varchar 50		
Espastica	Varchar 10		
Tipo estudio	Varchar 10		
ID_Persona	Number 10	x	Paciente(Cl,)
Talla	D_Talla		
Peso	D_Peso		

Atributo	Tipo de Dato Longitud	PK	FK
//Rodilla	//	//	//
Flex Rod der	D_FlexRod		
Flex Rod izq	D_FlexRod		
SLR	D_SLR		
Fuerza Flex der	dominio_fijo_t		
Fuerza Flex izq	dominio_fijo_t		
Fuerza Ext der	dominio_fijo_t		
Fuerza Ext izq	dominio_fijo_t		
Tono Flex der	dominio_fijo_t		
Tono Flex izq	dominio_fijo_t		
Tono Ext der	dominio_fijo_t		
Tono Ext izq	dominio_fijo_t		

Atributo	Tipo de Dato Longitud	PK	FK
Flex_cad_der	D_FlexCad		
Flex_cad_izq	D_FlexCad		
Ext_cad_der	D_ExtCad		
Ext_cad_izq	D_ExtCad		
Abd0_cad_izq	D_Abd0		
Abd0_cad_der	D_Abd0		
Abd9_cad_der	D_Abd9		
Abd9_cad_izq	D_Abd9		
Abd_cad_der	D_Abd		
Abd_cad_izq	D_Abd		

Atributo	Tipo de Dato Longitud	PK	FK
//	//	//	//
Selec Flex cad der	dominio_fijo_t		
Selec Flex cad izq	dominio_fijo_t		
Fuer Ext cad der	dominio_fijo_t		
Fuer Ext cad izq	dominio_fijo_t		
Fuer Abd cad der	dominio_fijo_t		
Fuer Abd cad izq	dominio_fijo_t		
Fuer Adu cad der	dominio_fijo_t		
Fuer Adu cad izq	dominio_fijo_t		
Tono mus Fle cad der	dominio_fijo_t		
Tono mus Fle cad izq	dominio_fijo_t		
Tono mus Ext cad der	dominio_fijo_t		
Tono mus Ext cad izq	dominio_fijo_t		
Tono mus Abd cad der	dominio_fijo_t		
Tono mus Abd cad izq	dominio_fijo_t		
Tono mus Adu cad der	dominio_fijo_t		
Tono mus Adu cad izq	dominio_fijo_t		
//////////	//	//	//
Ant Fem der	D_AntFem		
Ant Fem izq	D_AntFem		
Rot Int der	D_RotInt		
Rot Int izq	D_RotInt		
Rot Ext der	D_RotExt		
Rot Ext izq	D_RotExt		
Eje Bim der	D_EjeBim		
Eje Bim izq	D_EjeBim		
Ang pie der	D_AngPie		
Ang pie izq	D_AngPie		

Atributo	Tipo de Dato Longitud	PK	FK
//Tobillo	//	//	//
Dor Rod der9	D_DorRod9		
Dor Rod izq9	D_DorRod9		
Dor Rod der0	D_DorRod0		
Dor Rod izq0	D_DorRod0		
Plant Flex der	D_PlantFlex		
Plant Flex izq	D_PlantFlex		
Inver der	D_Inver		
Inver izq	D_Inver		
Ever der	D_Inver		
Ever izq	D_Inver		

Atributo	Tipo de Dato Longitud	PK	FK
//	//	//	//
Fue Mus Dorsi der	dominio_fijo_t		
Fue Mus Dorsi izq	dominio_fijo_t		
Fue Mus Plan der	dominio_fijo_t		
Fue Mus Plan izq	dominio_fijo_t		
Fue Mus Inv der	dominio_fijo_t		
Fue Mus Inv izq	dominio_fijo_t		
Fue Mus Ever der	dominio_fijo_t		
Fue Mus Ever izq	dominio_fijo_t		
Tono Flex Dor Izq	dominio_fijo_t		
Tono Flex Dor Der	dominio_fijo_t		
Tono Flex Plan Izq	dominio_fijo_t		
Tono Flex Plan Der	dominio_fijo_t		
Tono inv der	dominio_fijo_t		
Tono inv izq	dominio_fijo_t		
Tono ever izq	dominio_fijo_t		
Tono ever der	dominio_fijo_t		

Atributo	Tipo de Dato Longitud	PK	FK
//Pie	//	//	//
Antepie der	C_AnTPie		
Antepie izq	C_AnTPie		
Retropie der	C_ReTPie		
Retropie izq	C_ReTPie		
Red Anteder	C_RedAnt		
Red Anteizq	C_RedAnt		
Quimed der	C_QuiMed		
Quimed izq	C_QuiMed		
Redquie der	C_RedQui		
Redquie izq	C_RedQui		

Atributo	Tipo de Dato Longitud	PK	FK
//Medidas Biometricas	//	//	//
Distancia	Number 5		
Long der	Number 5		
Long izq)	Number 5		
DiBicond der	Number 5		
DiBicond izq	Number 5		
DiBima der	Number 5		
DiBima izq	Number 5		
//Reflejos	//	//	//
Osteot der	dominio_fijo_t		

Atributo	Tipo de Dato Longitud	PK	FK
Osteot_izq	dominio_fijo_t		
//Clonus	//	//	//
Clonus_der	C_CLO		
Clonus_izq	C_CLO		
//Observaciones	//	//	//
Características Marcha	Varchar 100		