

Apéndices

*¿Quién jamás ha echado de menos los falsos razonamientos
y varios caprichos para ser sabio,
sino los que buscan la vanidad en la sabiduría
y aman pensar de cualquier modo,
con tal que no piensen como los demás hombres?*

Ibid Juan Pablo Forner (1755-1797)

Apéndice I: Manual de Instalación de FuzzyCASE

Apéndice II: UML Difuso

Apéndice III: Modelos de Bases de Datos Difusas

Apéndice IV: Representación del Conocimiento Impreciso en un Base de Datos Relacional

Apéndice V: Museos Digitales en Internet: Modelo EER Difuso y Recuperación de Imágenes Basada en su Contenido

Apéndice VI Acrónimos

Apéndice I: Manual de Instalación de FuzzyCASE

I.1 Instalación de FuzzyCASE

Para comenzar a instalar FuzzyCASE se debe insertar el CD que se adjunta a esta tesis, luego se debe ir a la unidad correspondiente de CDROM (carpeta Mi PC). Ahí se encontrará, entre otras, la carpeta *Instaladores*. Se debe ingresar a esta carpeta y hacer doble click en el archivo SETUP.EXE (el que tiene de ícono un computador). La Figura I.1 muestra la pantalla de instalación de FuzzyCASE.

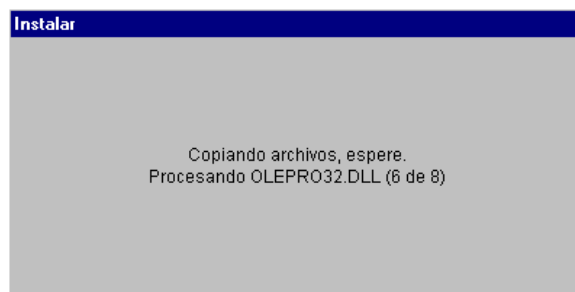


Figura I.1: Pantalla de inicio de instalación de FuzzyCASE.

Aquí, simplemente se debe esperar hasta que esta pantalla desaparezca. Continuando con la instalación aparecerá la pantalla que muestra la Figura I.2, donde solamente se debe clickear en el botón Aceptar.

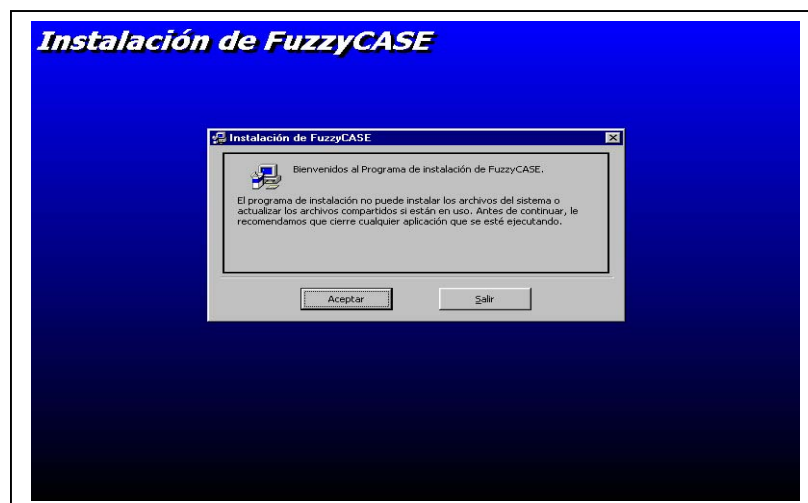


Figura I.2: Pantalla de instalación del FuzzyCASE.

Posteriormente aparecerá la pantalla de la Figura I.3, en la cual se debe clicar el botón que tiene un computador como ícono (indicado con un óvalo en la Figura I.3).

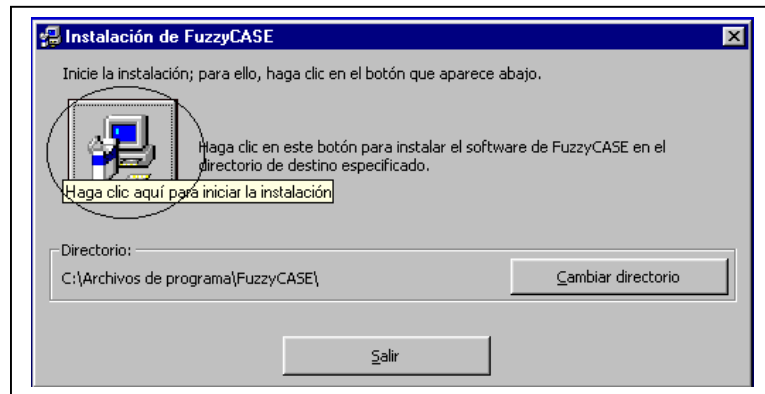


Figura I.3: Pantalla de creación de directorio c:\Archivo de programas\FuzzyCASE\.

Posteriormente, se debe transportar a la carpeta creada en la Figura I.3, un conjunto de programas propios de la instalación del FuzzyCASE. La Figura I.4 muestra la pantalla de esta tarea, donde simplemente, se debe clicar el botón continuar.

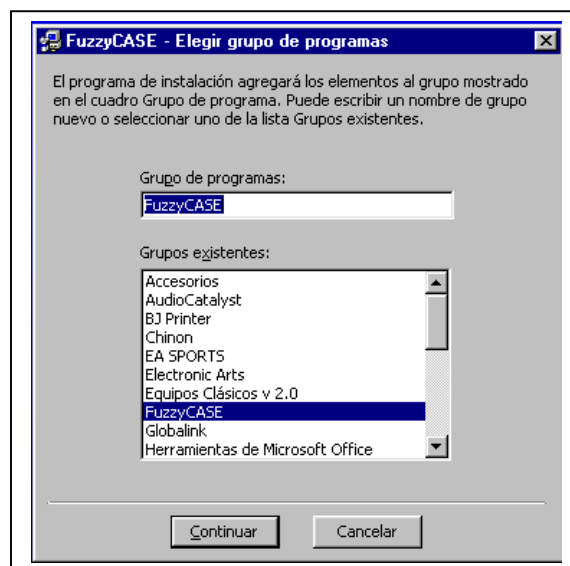


Figura I.4: Pantalla de carga de programas de la herramienta FuzzyCASE.

Una vez finalizado todos los pasos anteriores, se comenzará a instalar FuzzyCASE apareciendo la siguiente barra de progreso que muestra la Figura I.5.

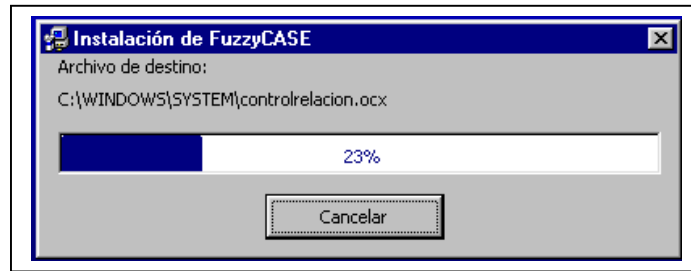


Figura I.5: Pantalla de progreso de instalación de FuzzyCASE.

En esta etapa se debe esperar a que termine (100%), una vez finalizada la instalación, esta pantalla desaparecerá.

Ahora bien, si en esta etapa aparece un mensaje preguntando si desea conservar un archivo ya existente, se deberá elegir la opción que indique que lo desea conservar.

Además, cuando FuzzyCASE se termine de instalar aparecerá este mensaje, aquí sólo debe clicar en el botón Aceptar para finalizar la instalación.

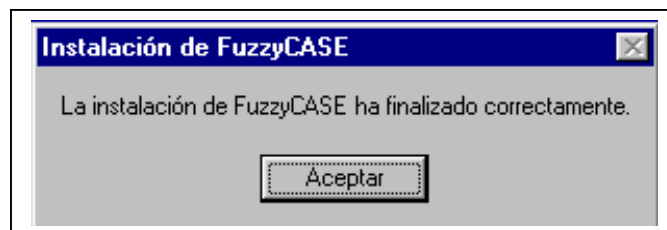


Figura I.6: Pantalla final de instalación FuzzyCASE.

I.2 Desinstalación

Para desinstalar FuzzyCase debe ir al botón inicio/Configuración/Panel de Control de Windows, tal como lo muestra la Figura I.7.

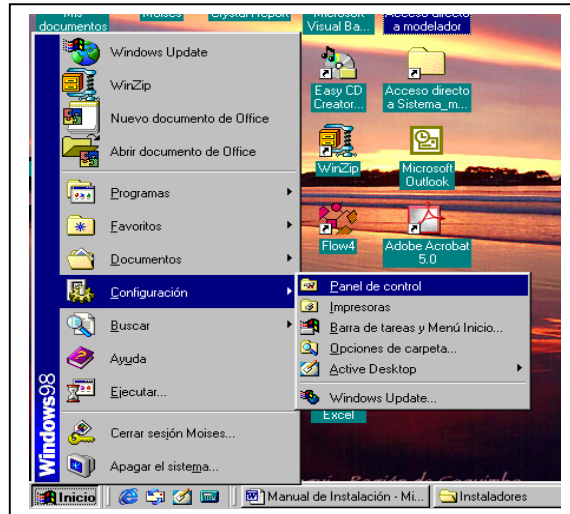
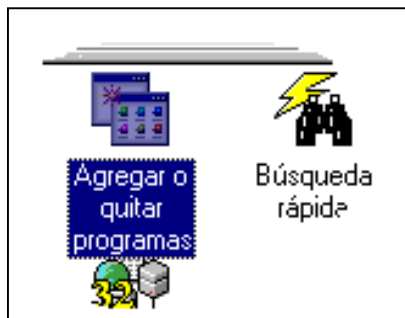


Figura I.7: Pantalla de desinstalación de FuzzyCASE.

Luego, en el Panel de Control se deberá hacer doble click en el ícono que dice: Agregar y quitar programas. La Figura I.8 muestra la pantalla del ícono.

a)



b)

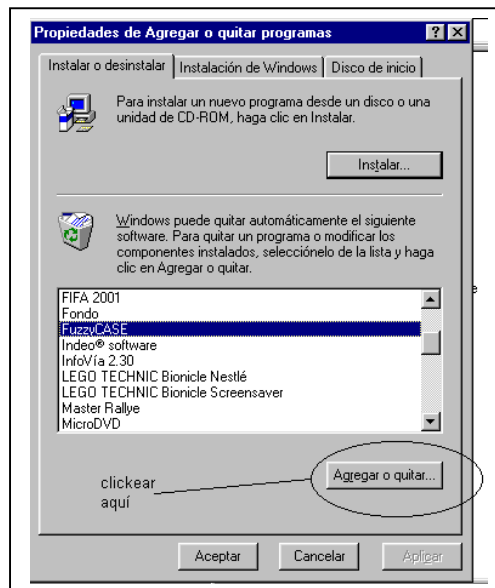


Figura I.8: Pantallas de agregar o quitar programas.

Al hacer click en la pantalla a) de la Figura I.8, aparecerá la pantalla b) de la Figura I.8. En esta pantalla, y tal cual como se indica, se debe seleccionar en instalar o desinstalar el programa FuzzyCASE (en la imagen de arriba esta seleccionado en azul de la Figura I.8 b)). Después de haberlo seleccionado se debe hacer click en el botón Agregar o quitar. la Figura I.9 muestra la pantalla donde debe asegurarse que desea desinstalar FuzzyCASE opción “Sí” u opción “No” si no lo desea.

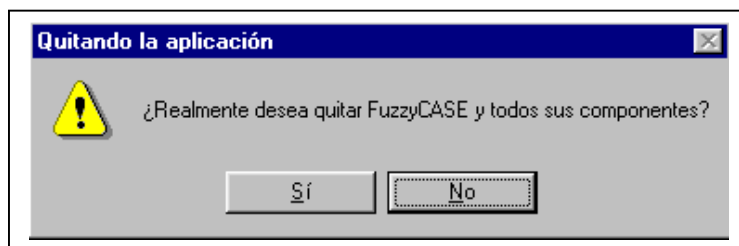


Figura I.9: Pantalla de compromiso de desinstalación.

Para finalizar la desinstalación, aparecerá una barra de progreso al llegar al 100%, desaparecerá esta barra (por si sola). El proceso se cierra cuando aparecerá la pantalla de la Figura I.10.

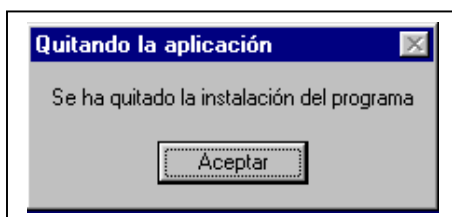


Figura I.10: Pantalla final de desinstalación.

Aquí se debe clicar en el botón Aceptar. Con esto finaliza la desinstalación.

Nota: En el caso que apareciere algún mensaje preguntando si desea quitar algún archivo se puede hacer un click en *Quitar* y revisar que archivo es, o en su defecto hacer click en *Quitar Todos*. De todos modos se recomienda hacer click en *Quitar Todos*.

I.3 Ejemplos del CD de FuzzyCASE

En el CD que adjunta esta tesis, contiene una carpeta de *Ejemplos* que contienen algunos ejemplos de esquemas que han sido descritos en la tesis. Es muy importante que para poder usar estos archivos de ejemplos se deben seguir los siguientes pasos:

1. Copiar la carpeta de ejemplo al disco duro.
2. Desactivar el modo de la carpeta de sólo lectura a modificar (véase Figura I.11).

Una vez copiados los archivos al disco duro se deben cambiar las propiedades de los archivos. Esto se hace haciendo click sobre el archivo con el botón derecho del Mouse. En el menú que aparece se debe ir a la opción *Propiedades*. La Figura I.11 muestra la pantalla con la opción de modificación de ejemplos.

Estos pasos se deberán repetir con cada uno de los archivos de ejemplos. Es importante señalar que: “el no hacer estos pasos para usar los ejemplos puede hacer que el programa no funcione correctamente”.

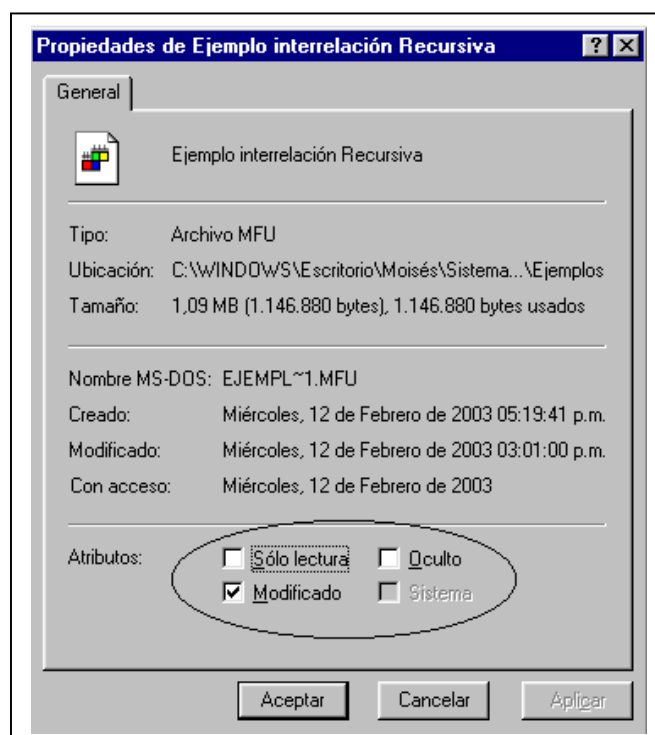


Figura I.11: Pantalla de cambio de atributos de archivo ejemplo.

Para ejecutar FuzzyCASE se debe ir a la *Barra Inicio*, programas, FuzzyCASE, opción FuzzyCASE, de esta forma se cargará el software de la herramienta gráfica. Al ingresar a la barra principal opción archivo abrir carpeta ejemplos, como lo muestra Figura I.12, en pantalla superior parte del caso de la agencia inmobiliaria, pantalla inferior subclases compartida de caso vehículo registrado.

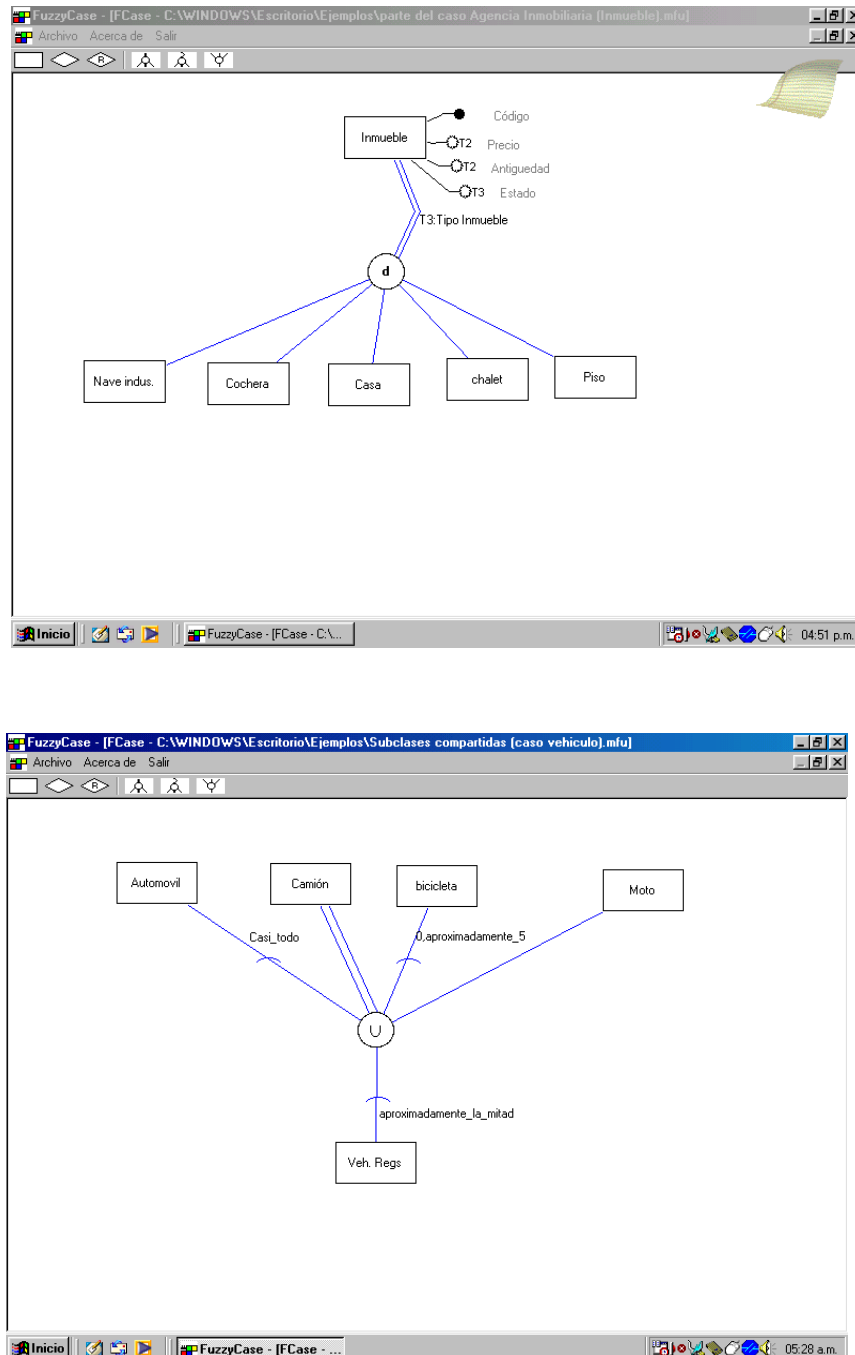


Figura I.12: Pantallas de FuzzyCASE con ejemplos.

Apéndice II: Diagrama de Clases en UML Difuso

El UML (Lenguaje Unificado de Modelado), es un lenguaje de modelado que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software (Fowler y Scott, 1999). Captura decisiones y conocimiento sobre los sistemas que se deben construir, su objetivo es lograr que, además de describir con cierto grado de formalismo, tales sistemas, puedan ser entendidos por los usuarios de aquello que se modela.

En este apartado, se muestra una extensión del modelado UML en especial atributos, y cardinalidades, la notación obtenida es llamada FuzzyUML. Cabe destacar que la implementación de la notación propuesta, se ha implementado en la herramienta Rational Rose.

II.1 Atributos difusos en FuzzyUML

Los atributos difusos definidos en el apartado 3.2.4 se extienden ahora en el lenguaje de modelamiento UML con el uso de un patrón de diseño para “Atributos Difusos” que permite incluir atributos difusos en las clases en sus diseños. A esta extensión la llamaremos FuzzyUML.

Rumbaugh et al., (1999) definen un atributo en UML como la descripción de una ranura con el nombre de un tipo especificado en una clase; cada objeto de la clase tiene un valor independiente para el atributo. Un atributo se representa mediante una cadena de texto que puede dividirse en varias propiedades su notación es <<estereotipo>> visibilidad nombre multiplicidad: tipo = valor-inicial {cadena de propiedades}.

También definen una clase como un descriptor de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y comportamiento. Una clase representa un concepto dentro del sistema que se esta modelando. Una clase debe identificar el nombre de la clase atributos y operaciones.

Por otro lado, definen un <<estereotipo>> como una nueva clase de elemento del modelo definida dentro del modelo y basada en alguna clase existente de elementos del modelo, Los

sistemas pueden extender la semántica pero no la estructura de clases preexistente del metamodelo.

La propuesta para extender los atributos de un modelo de datos UML a un tratamiento impreciso, en primera instancia, la representamos con el uso de clases y estereotipos.

El enfoque propuesto se basa en la definición de los estereotipos «FuzzyT1», «FuzzyT2» y «FuzzyT3» para cada uno de los atributos difusos del Tipo 1, Tipo 2 y Tipo 3, respectivamente. Los estereotipos mencionados corresponden a clases que incorporan los métodos y atributos necesarios para manejar las propiedades difusas de estos datos. Los tipos de datos difusos extendidos en FuzzyUML son:

- La clase «FuzzyT1» tiene asociado una lista de etiquetas lingüísticas, de modo de posibilitar el manejo de los atributos difusos Tipo 1. Véase Figura II.1.
- La clase «FuzzyT2» tiene asociada una lista de trapecios que tienen asociada una etiqueta lingüística. El tipo de dato de los atributos a, b, c y d debe asignarse según el dominio del atributo difuso Tipo 2 que se utilizará.
- La clase «FuzzyT3» tiene asociada una matriz que almacena los grados de similitud (gs) que se le atribuyen a los pares de etiquetas lingüísticas en consideración. No siempre la relación de similitud es simétrica, por lo que se diferencié el orden de las etiquetas en la relación (posición 1 y posición 2).

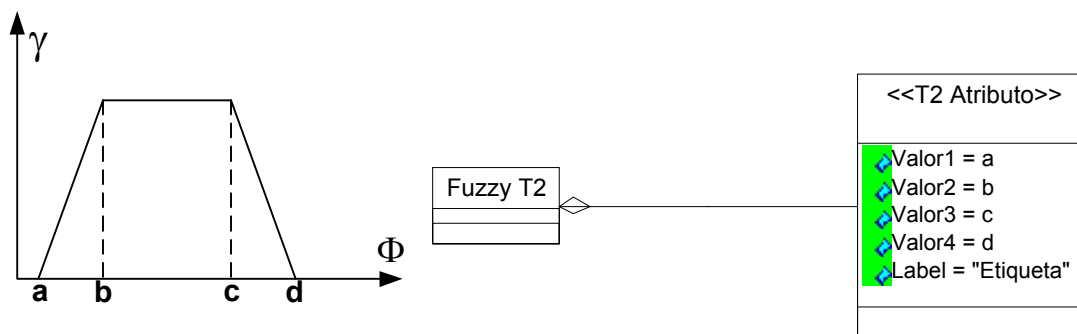


Figura II.1: Representación de un atributo difuso Tipo 2 en FuzzyUML.

Para aplicar el estereotipo a un caso concreto, se debe hacer uso de la relación de especialización, y generar una clase del tipo de atributo requerido, y asignarle el tipo de dato correspondiente. Este proceso se realiza en la “vista de definición difusa”. En la vista estática del problema en cuestión se usará la clase como dominio del atributo difuso correspondiente, el cuál además será identificado con la etiqueta del estereotipo correspondiente.

Rumbaugh et. al. (2000) definen un patrón que denota una colaboración parametrizada que representa un conjunto de clasificadores, relaciones y comportamiento parametrizados, que se pueden aplicar a múltiples situaciones enlazando elementos del modelo (clases normales) con los roles del patrón. En una planilla de colaboración.

En un patrón la colaboración es la solución para las distintas situaciones de la especialización de cada uno de los tipos de atributos, ya sea éste: Tipo 1 que puede definir varias etiquetas con cardinalidad 1..*; Tipo 2 que puede definir uno a más trapecios cardinalidad 1..*; y Tipo 3 que genera una función de similitud de uno a más valores con cardinalidad 1..*. La Figura II.2 representa el uso de patrones de atributos difusos Tipo 1, 2 y 3 en FuzzyUML.

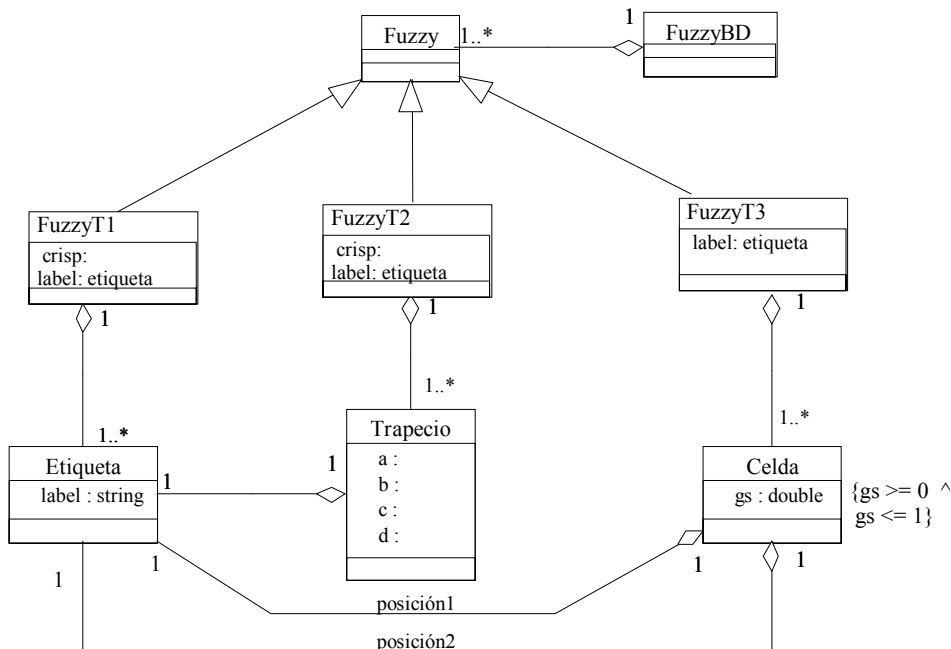


Figura II.2: Vista estática de patrón en atributos difusos.

El atributo crisp presente en FuzzyT1 y FuzzyT2 puede admitir valores precisos del tipo de dato especificado por el diseñador, además de los valores Unknown, Undefined y Null. El atributo etiqueta puede tomar como valor cualquiera de las etiquetas que están asociadas a la clase (que determina de ese modo el dominio de etiquetas posibles). Puede tomar el valor nulo cuando crisp tiene valor.

Los atributos difusos Tipo 2 tienen asociado uno o más trapecios como éste. A cada valor entre a y d se le puede asociar un grado de pertenencia al label o etiqueta.

II.2 Métodos en Atributos Difusos

Rumbaugh et. al. (1999) definen un método como una implementación de una operación. También, especifican el algoritmo o procedimiento que da lugar a los resultados de una operación. Los métodos para cada clase del patrón de la Figura II.2 se describen informalmente a continuación.

Clase Fuzzy

- **Asignar etiqueta:** Asigna una etiqueta entre las definidas en la clase, a una instancia específica.
- **Retornar Grado de Pertenencia:** Retorna el grado de pertenencia de la instancia a una etiqueta lingüística (se especializa para cada subclase).

Clase: FuzzyT1

- **Retornar crisp:** Indica cual es el valor exacto del atributo.

Clase: FuzzyT2

- **Retornar Grado de Pertenencia:** Dado un valor crisp, se evalúa en cada trapecio el grado de pertenencia a cada etiqueta lingüística.

- **Retornar rango de valores:** Dada una etiqueta lingüística y un grado de pertenencia, retorna el rango de valores que califica.

Clase FuzzyT3

- **Devolver grado de similitud** entre 2 etiquetas dadas, retorna el grado de similitud.

Nótese que se pueden definir otros métodos o cuantos el usuario estime conveniente para cada caso, esto dependerá de los requerimientos de lo que se desee modelar.

Ejemplo II.1: Para un atributo difuso Tipo 2, considérese que se desea modelar la clase personas utilizando como atributo difuso la Edad. Se utilizarán las etiquetas “Infante”, “Joven” y “Adulto”. Los valores de los atributos a, b, c y d correspondientes a cada uno de los trapecios (asociados a cada una de las tres etiquetas lingüísticas) se definen en la instanciación de la vista de definición difusa.

En la vista estática del diseño en cuestión, se define el atributo difuso dentro de la clase, como se muestra a continuación.

Persona
«fuzzyT2» Edad : Vejez

Figura II.3: Vista estática con inclusión de un atributo difuso.

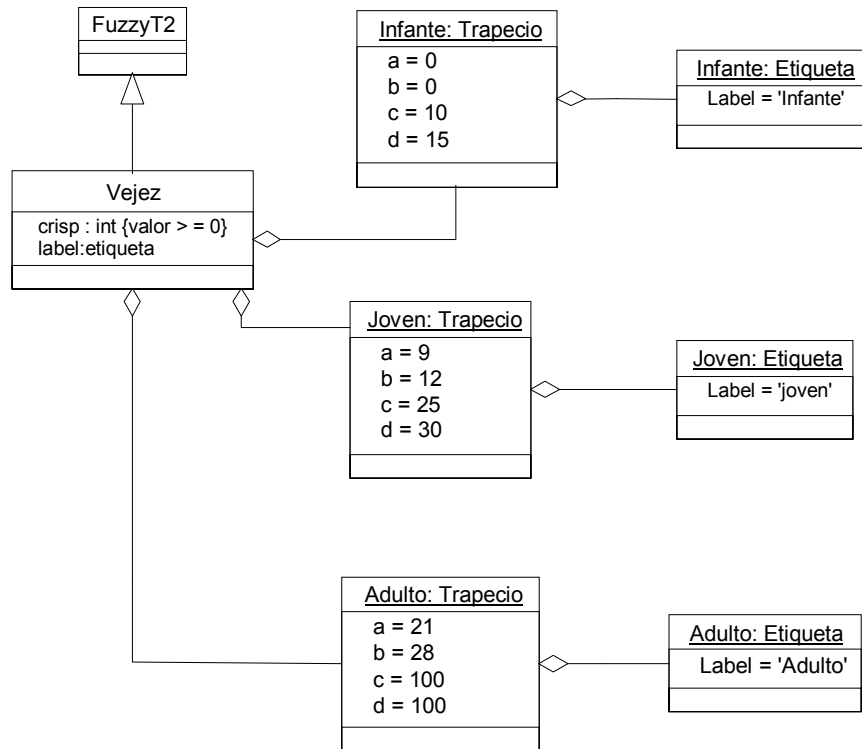


Figura II.4: Vista de definición difusa para el caso propuesto.

Ejemplo II.2: Para el atributo difuso Tipo 3. se ha elegido el ejemplo, color del pelo de las personas. Definiendo tres colores: rubio, castaño y pelirrojo como etiquetas lingüíticas. Se ha establecido los grados de similaridad entre estas etiquetas según se muestra en la Tabla II.1 y el mapeo de esta tabla al diseño de diagrama de clase en UML se realiza según esta tabla.

Color de pelo	Rubio	Castaño	Pelirrojo
Rubio	1	0.1	0.5
Castaño	0.1	1	0.7
Pelirrojo	0.5	0.7	1

Tabla II.1: Grados de similaridad de atributo color de pelo Rubio, Castaño y Pelirrojo.

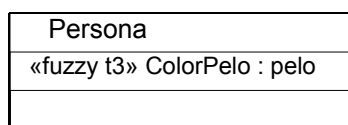


Figura II.5: Vista estática con uso del estereotipo FuzzyT3 de atributo color del pelo.

Para el caso del atributo difuso Tipos 3 del ejemplo II.1, la Figura II.6 muestra el modelado de la Tabla II.1, considerando el FuzzyUML con el uso de patrones.

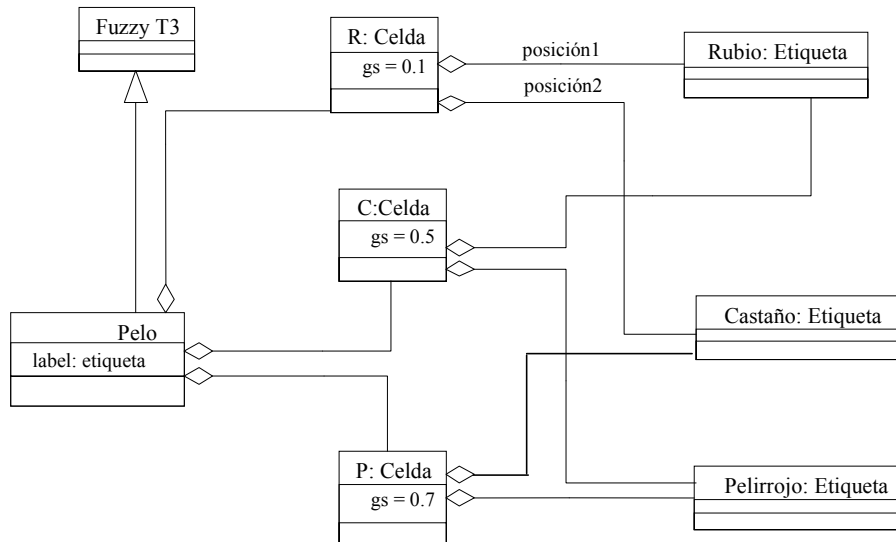


Figura II.6: Vista de definición difusa para atributos difusos Tipo 3.

II.3 Métodos en Atributos Difusos para Consultas

En este apartado nos enfocaremos sólo en la definición de métodos para consultar la base de datos difusa según los comparadores de la Tabla II.2, como una clase FuzzyBD. Para ello usaremos las definiciones provistas en: Galindo (1999), Dubois y Prade (1998) y Medina (1994), con respecto a comparadores difusos y su uso en consultas difusas. La definición de los mecanismos de evaluación de las sentencias de la forma (Atributo Operador Valor) debe hacerse a priori y constituye un nivel de abstracción inferior en el diseño. A continuación definiremos los comparadores utilizados en los métodos de consultas.

Comparadores Difusos.

Además de los comparadores comunes (=,<,>, etc.), un servidor FSQL incluye los comparadores difusos de la Tabla II.2. Al igual que el SQL los comparadores difusos pueden comparar dos columnas del mismo tipo o de tipos compatibles. Éstos pueden ser clasificados en comparadores de posibilidad o necesidad.

Los *comparadores de posibilidad* (véase Tabla II.2) son más generales (menos restrictivos) que los de necesidad. Por tanto, los *comparadores de necesidad* (véase Tabla II.2) recuperan menos instancias y estas instancias cumplirán necesariamente con las condiciones impuestas en la consulta.

Comparador Posibilidad	Comparador Necesidad	Significado
FEQ	NFEQ	Igual
FGT	NFGT	Mayor
FGEQ	NFGEQ	Mayor o Igual
FLT	NFLT	Menor
FLEQ	NFLEQ	Menor o Igual
MGT	NMGT	Mucho Mayor
MLT	NMLT	Mucho Menor

Tabla II.2: Comparadores de POSIBILIDAD y NECESIDAD.

Todas las operaciones definidas a continuación son métodos de la clase FuzzyBD y actúan sobre las clases FuzzyT1 y FuzzyT2. Sin embargo, el caso de FuzzyT3 sólo se utiliza el comparador FEQ, ya que no tienen un referencial ordenado asociado, lo que indica que no puede usarse comparadores $>$ y $<$. La restricción de posibilidad y necesidad queda definida por el conjunto: ClaseD (que define la clase que utilizará), Atributo (que tendrá el valor crisp de atributo), Operador (que establece el operados usado), Valor (valor del grado devuelto con umbral de cumplimiento γ), quedando como:

- Restricción de Posibilidad (ClaseD, Atributo, Operador, Valor): Este método selecciona todos los objetos de la clase ClaseD que satisfacen (Atributo Operador Valor) donde Operador pertenece a {FEQ, FGT, FGEQ, FLT, FLEQ, MGT, MLT} y Valor pertenece al dominio de Atributo.
- Restricción de Necesidad (ClaseD, Atributo, Operador, Valor): Este método selecciona todos los objetos de la clase ClaseD que satisfacen (Atributo Operador Valor) donde Operador pertenece a {NFEQ, NFGT, NFGEQ, NFLT, NFLEQ, NMGT, NMLT} y Valor pertenece al dominio de Atributo.

Para cada uno de los atributos definidos podemos ejemplificar las consultas de la siguiente forma:

- Consulta FuzzyT1 (ClaseT1, Atributo, Label): Este método selecciona todos los objetos de la clase difusa Tipo 1 ClaseT1, tal que el valor del atributo del tipo Atributo esté asociado a la etiqueta Label.
- Consulta FuzzyT2 (ClaseT2, Atributo, Label, gp): Este método selecciona todos los objetos de la clase difusa Tipo 2 ClaseT2, tal que el valor del atributo del tipo Atributo esté asociado al rótulo Label con un grado de pertenencia mayor a gp.
- Consulta FuzzyT3 (ClaseT3, Atributo, Label, gs): Este método selecciona todos los objetos de la clase difusa Tipo 3 ClaseT3, tal que el atributo del tipo Atributo tenga un grado de similitud en relación al rótulo Label mayor a gs.

Hasta aquí en lo que se refiere a la extensión en UML, se ha realizado una propuesta para expresar atributos que sus dominios sean imprecisos y la forma de representarlos en un modelo conceptual como UML. Las representaciones expuestas tienen un novedoso significado y aportan gran expresividad al modelo conceptual.

En cuanto a la cardinalidad, generalización y especialización se pueden utilizar los cuantificadores difusos al igual que las propuestas para el modelo FuzzyEER.

Apéndice III: Modelos de Bases de Datos Difusas

El primer intento de representar datos imprecisos en una base de datos fue la introducción de valores nulos, NULL, por parte de Codd en 1976, este modelo no utilizaba la teoría de conjuntos difusos. Un valor NULL en un atributo indica que dicho valor es cualquiera de entre todos los que estuvieran incluidos en el dominio de dicho atributo. Cualquier comparación con un valor NULL, origina un resultado que no es ni verdad, ni falsedad, llamado “quizás” (o desconocido, unknown en SQL). Después de esta aproximación de Codd, han surgido algunos modelos básicos de Bases de Datos Difusas, que mostramos a continuación cuyas definiciones han sido extraídas de Pons (1996) y Galindo (1999).

Para *el tratamiento de la información precisa e incierta* existen tres enfoques dentro de los cuales se podrían clasificar los distintos modelos difusos, éstos son:

- Modelo Relacional Difuso.
- Modelo de Relaciones de Similitud.
- Modelo Relacional Posibilístico.

III.1 Modelo Relacional Difuso

El modelo básico de base de datos relacional difuso se considera la forma más simple y consiste en añadir un *grado*, normalmente en el intervalo $[0,1]$, a cada instancia. Esto permite mantener la homogeneidad de los datos de la base de datos. Sin embargo, la semántica que se le asigna a ese grado será la que determine su utilidad y, por tanto, esta semántica será utilizada en los procesos de consulta.

Este grado puede ser sobre *un valor*, o así como también pueden ser aplicados *a una instancia (tupla) o a un conjunto de valores* en otros modelos. El significado de estos grados puede ser variado, es por esta razón que a partir de éste será el tratamiento de los datos. Entre los significados de los datos se puede mencionar: *grado de cumplimiento, grado de pertenencia, grado de incertidumbre, etc.*

El principal inconveniente de estos modelos difusos es que no permiten representar la información imprecisa que se tiene sobre algún atributo particular de alguna entidad concreta (como los valores “joven” o “viejo” para un atributo “edad”). Además, el carácter difuso es

asignado de forma global a cada instancia sin que se pueda determinar la aportación difusa particular de cada uno de los atributos que la constituyen.

Formalmente, una BDRD está constituida por un conjunto de relaciones (representado en la igualdad a)) en donde cada una de ellas es una relación difusa caracterizada por una función de pertenencia (indicado en b)):

- a) $BDRD = \{ R_1, R_2, \dots, R_n \}$
- b) $\mu_{R_i}: U_1 \times U_2 \times \dots \times U_n \rightarrow [0, 1]$

Siendo μ_{R_i} una función de pertenencia, R_i la relación i-ésima, U_i el dominio del atributo i-ésimo de la relación y \times el producto cartesiano.

III.2 Modelo de Relaciones de Similitud

Es el primer modelo que utiliza relaciones de similitud en el modelo relacional fue propuesto por Buckles y Petry (1982).

En éste una *Relación Difusa* es definida como un subconjunto del producto cartesiano siguiente: $P(D_1) \times \dots \times P(D_m)$, donde $P(D_i)$ representa el conjunto de las partes de un dominio D_i , que incluye a todos los subconjuntos que pueden considerarse dentro del dominio D_i (con cualquier número de elementos).

La forma inicial de representar y manejar la imprecisión en este modelo, era a través de relaciones de similitud sobre los dominios permitidos para los atributos, éstos eran los siguientes:

- Conjunto finito de escalares.
- Conjunto finito de números.

Más tarde con el modelo propuesto por Buckles y Petry, se incorpora un nuevo dominio para este modelo, que es un:

- Conjunto de números difusos.

Dicho modelo propone una extensión para un tratamiento de este tipo de dominio.

Aquí las clases de equivalencia sobre un dominio se construyen a partir de *las Relaciones de Similitud*, en la cual los valores tomados por esta relación deberán ser suministrados por el usuario, normalmente estos valores de similitud están normalizados en el intervalo $[0,1]$, correspondiendo el 0 al significado “*totalmente diferentes*” y el 1 al significado “*totalmente parecidos*” y ofrecen la medida en que los valores del dominio se parecen según la apreciación del usuario, y de un *umbral de similitud* mínimo en la consulta.

Entendiendo por *umbral de similitud*, aquel valor entre 0 y 1 establecido para ejercer algún control sobre la precisión con que se satisface cada una de las condiciones de la consulta.

El proceso de consulta de un usuario sería similar al de una BDRD debido a que el usuario pregunta por aquellas instancias que satisfacen una determinada condición con unos determinados umbrales de similitud. La diferencia existe que el sistema toma dicha condición y a partir de las relaciones de similitud definidas agrupa en clases de equivalencia las instancias de la relación de partida. Finalmente opera sobre las clases de equivalencia formadas con los operadores tradicionales del álgebra relacional.

III.3 Modelo Relacional Posibilístico

Bajo la representación de los valores de dominios representados por distribuciones de posibilidad y la definición de una BDRD existen muchos modelos definidos, donde la diferencia entre ellos radica principalmente en los siguientes aspectos:

- Representación de las distribuciones de posibilidad y manejo de información desconocida, indefinida y nula.
- Planteamiento de la consulta.
- Elección de los operadores de selección con los que evaluar las condiciones impuestas en una consulta.

Estos modelos son:

1. Modelo de Prade-Testemale.
2. Modelo de Umano-Fukami.
3. Modelo de Zemankova-Kaendel.

III.4 Modelo Prade-Testemale

Prade y Testemale publicaron un modelo de BDRD que permite incorporar lo que denominan datos *incompletos* o *inciertos* en el ámbito de la Teoría de la Posibilidad (Prade y Testemale, 1982).

Se considera un atributo A cuyo dominio es D . Todo el conocimiento disponible acerca del valor que toma A para un objeto x puede ser representado mediante una *distribución de posibilidad* $\pi_{A(x)}$ sobre $D \cup \{e\}$, donde e es un elemento especial que denota el caso en que A no se aplica a x . En otras palabras, $\pi_{A(x)}$ es una aplicación que va de $D \cup \{e\}$ al intervalo $[0,1]$. A partir de esta formulación, se pueden representar todos los tipos de valores adoptados por este modelo.

En todos los modelos posibilísticos hay que tener en cuenta que si se tiene que, para un $d \in D$, se cumple que $\pi_{A(x)}(d) = 1$, esto sólo indica que el valor d es completamente posible para $A(x)$, y no que el valor d sea cierto para $A(x)$, a menos que ese sea el único valor posible, es decir, $\pi_{A(x)}(d') = 0$, para todo $d' \neq d$. La Tabla III.1 muestra la información y representación de este modelo.

Información	Representación
Sabemos exactamente el dato y éste es puro: c	$\pi_{A(x)}(e) = 0$ $\pi_{A(x)}(c) = 1$ $\pi_{A(x)}(d) = 0, \forall d \in D, d \neq c$
Desconocida pero aplicable	$\pi_{A(x)}(e) = 0$ $\pi_{A(x)}(d) = 1, \forall d \in D$
No aplicable o sin sentido	$\pi_{A(x)}(e) = 1$ $\pi_{A(x)}(d) = 0, \forall d \in D$
Ignorancia Total	$\pi_{A(x)}(d) = 1, \forall d \in D \cup \{e\}$
Rango $[m, n]$	$\pi_{A(x)}(e) = 0$ $\pi_{A(x)}(d) = 1$ si $d \in [m, n] \subseteq D$ $\pi_{A(x)}(d) = 0$ en otro caso
La información disponible es una distribución de posibilidad μ_a	$\pi_{A(x)}(e) = 0$ $\pi_{A(x)}(d) = \mu_a(d) \forall d \in D$
La posibilidad de que no sea aplicable es λ y, en caso de que sea aplicable el dato es μ_a	$\pi_{A(x)}(e) = \lambda$ $\pi_{A(x)}(d) = \mu_a(d) \forall d \in D$

Tabla III.1: Representación de información de Prade-Testemale.

III.5 Modelo de Umano-Fukami

Esta propuesta utiliza las distribuciones de posibilidad para modelar el conocimiento sobre la información. En este modelo la información no aplicable, que entienden que puede ser modelada por una distribución de posibilidad sobre el dominio considerado, en la que cada valor de dominio aparece con un valor de posibilidad igual a 0. Es decir, si D es el universo de discurso de $A(x)$ y $\pi_{A(x)}(d)$ representa la posibilidad de que $A(x)$ tome el valor u en U , entonces para los valores desconocidos y aplicables, que denomina *unknown*, emplea la representación siguiente:

$$Unknown = \pi_{A(x)}(d) = 1 \quad \forall d \in D$$

Para los valores *no aplicables* existe un caso especial de distribución de posibilidad denominado *undefined*, el cual se representa:

$$Undefined = \pi_{A(x)}(d) = 0 \quad \forall d \in D$$

Para representar la situación en la que no se conoce incluso si una *ausencia* de información es *aplicable* o *no aplicable*, emplean un valor especial que denominan *Null*, indicado a continuación:

$$Null = \{1/Unknown, 1/Undefined\}$$

Para el resto de los casos de información imprecisa adopta una representación similar al modelo anterior.

Además, cada instancia de una relación en el modelo tiene asociada una distribución de posibilidad en el intervalo $[0,1]$, de forma que indica el grado de pertenencia de esa instancia a esa relación. O sea, una relación difusa R , de m atributos, queda definida como la función de pertenencia siguiente:

$$\mu_R: P(U_1) \times P(U_2) \times \dots \times P(U_m) \rightarrow P([0,1])$$

donde el símbolo x denota el producto cartesiano, $P(U_j)$ con $j=1, 2, \dots, m$ es la colección de todas las distribuciones de posibilidad en el universo de discurso U_j del j -ésimo atributo de R .

La función μ_R asocia a cada instancia de la relación R un valor $P([0,1])$, que es el conjunto de todas las distribuciones de posibilidad en el intervalo $[0,1]$ y que será visto como un grado de pertenencia de esa instancia a R .

Finalmente, en el proceso de consulta, expresada en términos difusos o precisos, el modelo resuelve la consulta dividiendo el conjunto de las instancia implicadas en la relación en tres subconjuntos, en donde el primer subconjunto contiene las instancia que satisfacen completamente la consulta, el segundo subconjunto agrupa a aquellas instancia que posiblemente satisfagan la consulta, y por último el tercer subconjunto está compuesto por aquellas instancia que no satisfacen la consulta. La Tabla III.2 muestra la información y representación de este modelo.

Información	Representación
Sabemos exactamente el dato y éste es puro: c	$\pi_{A(x)}(d) = \{1 / c\}$
Desconocida pero aplicable	Unknown
No aplicable o sin sentido	Undefined
Ignorancia Total	Null
Rango $[m, n]$	$\pi_{A(x)}(d) = 1$ si $d \in [m, n] \subseteq D$ $\pi_{A(x)}(d) = 0$ en otro caso
La información disponible es una distribución de posibilidad μ_a	$\pi_{A(x)}(d) = \mu_a(d) \quad \forall d \in D$
La posibilidad de que no sea aplicable es λ y, en caso de que sea aplicable el dato es μ_a	No representable

Tabla III.2: Representación de información de Umano-Fukami.

III.6 Modelo Zemankova-Kandel

Este modelo se compone de tres partes:

- Una base de datos de valores, donde se organizan los datos en forma similar a los modelos posibilísticos anteriores.
- Una base de datos explicativa, donde se almacenan las definiciones para los subconjuntos difusos y relaciones difusas utilizados.
- Un conjunto de reglas de traducción, para el manejo de adjetivos y modificadores.

El planteamiento de la consulta se hace en forma similar al modelo de Umano y Fukami, salvo que la medida de posibilidad que se emplea para encontrar la compatibilidad del subconjunto difuso F de la condición, con el valor del atributo A para cada instancia en la relación, viene dada por:

$$P_A(F) = \sup_{u \in D} \{ \mu_F(u) * \pi(u) \}$$

Y la medida de certeza está dada por:

$$C_A(F) = \max_{u \in D} \{ 0, \inf \{ \mu_F(u) * \pi_A(u) \} \}$$

ésta es usada en lugar de la necesidad del modelo Prade-Testemale. Sin embargo, la interpretación del grado de certeza no está clara y no hay relación entre la posibilidad y la certeza, como sí la hay entre posibilidad y necesidad: $N(X) = 1 - P(\neg X)$.

El resultado de una consulta se presenta en forma de relaciones difusas las cuales contienen dos campos en los que se recogen los valores de posibilidad y certeza que presenta cada instancia para la consulta dada. Sobre estas relaciones se pueden establecer unos umbrales mínimos a satisfacer para las instancia que se recuperen.

Al imponer condiciones a la hora de realizar una selección, se parte de una relación de similaridad definida sobre $D \times D$, a partir de la cual se construye cualquier otra relación de comparación.

III.7 Modelo GEFRED

El modelo GEFRED se encuentra en Medina (1994). Este modelo es una síntesis ecléctica de algunos de los modelos anteriormente presentados. En forma especial hace referencia a dominios difusos generalizados, siendo este un modelo posibilístico, por lo que en los dominios admite distribución de posibilidad, pero también incluye el caso en el que el dominio subyacente no sea numérico, sino escalares de cualquier tipo, además incluye los valores Unknown, Undefined y Null con el mismo sentido de Umano y Fukami.

El modelo GEFRED incluye relaciones difusas generalizadas, que son relaciones cuyos atributos tienen un dominio difuso generalizado considerando los siguientes casos:

- A cada atributo A_j es posible asociarle un “atributo de compatibilidad” C_j donde almacenar un grado considerando:
- Que el grado se obtiene como consecuencia de los procesos de manipulación de los datos de esa relación.

- Expresa el grado con el que ese valor ha satisfecho la operación realizada sobre él.

A su vez, las relaciones se componen de:

- Cabeza H, que es el nombre de cada uno de los n atributos, sus dominios y sus atributos de compatibilidad (opcional).
- Cuerpo B, que incluye los valores de m tuplas:

La siguiente fórmula muestra los componentes de la relación

$$R = \begin{cases} H = \{(A_1:D_1[C_1], \dots, A_n:D_n[C_n])\} \\ B = \{(A_1:d_{i1}[c_{i1}], \dots, A_n:d_{in}[c_{in}])\} \text{ con } i=1, \dots, m \end{cases}$$

El modelo GEFRED define comparadores difusos, que son un tipo de comparador general basado en cualquier comparador clásico existente ($>$, $<$, $=$, etc.), pero no concreta la definición de cada uno. El único requisito que establece es que el comparador difuso debe respetar los resultados de los comparadores clásicos cuando se compara una distribución de posibilidad que expresan valores *crisp* (como $1/x$ con x perteneciente a X). Por otro lado, GEFRED, incluye las componentes del álgebra difusa.

Apéndice IV: Representación del conocimiento impreciso en un Base de Datos Relacional

IV.1 Implementación de atributos difusos

La propuesta aquí presentada se basa en GEFRED (Medina, 1994; Galindo 1999), donde, existen tres tipos de atributos que pueden tener de tratamiento impreciso. La clasificación adoptada se basa en criterios de representación y tratamiento de los datos “imprecisos”: Se clasifican según el tipo del dominio que les subyace y por si permite el almacenamiento de información imprecisa o sólo permite el tratamiento impreciso de datos sin imprecisión:

Atributos difuso Tipo 1: Este tipo de atributo recibe una representación igual que los dato precisos. Sin embargo, cuenta con información en la FMB (Base de Metaconocimiento Difuso), donde se almacenarán las etiquetas, también recogerá información acerca de la naturaleza de estos atributos. Por tanto, son atributos clásicos que admiten el tratamiento difuso y por tanto podremos efectuar consultas (flexibles) sobre ellos, aunque no permita almacenar valores difusos.

Atributos difuso Tipo 2: Este tipo de atributo permite almacenar información imprecisa sobre dominios ordenados. En la Tabla IV.1 mostramos el sistema utilizado para representar a los atributos difusos Tipo 2. Así, vemos que un atributo difuso Tipos 2, llamado por ejemplo F, está compuesto, de hecho, por cinco atributos clásicos:

FT: Almacena el *tipo de valor* que corresponde al dato que queremos almacenar, indicando su representación: Según lo visto, puede ser: UNKNOWN (0), UNDEFINED (1), NULL (2), CRISP (3), LABEL (4), INTERVALO (5), APROXIMADAMENTE (6) y TRAPEZOIDAL (7).

F1, F2, F3 y F4: Los atributos cuyos nombres se forma añadiendo los números 1, 2, 3 y 4 al nombre del atributo que almacena la *descripción* de los parámetros que definen el dato y dependen del tipo de valor (FT) al que pertenezca.

UNKNOWN, UNDEFINED, NULL: Estos tres valores no necesitan ningún parámetro, por lo que todos ellos permanecen Null (entendiendo este valor como el NULL del SGBD anfitrión y no como el Null del valor difuso, que no deben confundirse).

- **CRISP:** Un valor de tipo Crisp, necesita tan sólo un parámetro, F1, en el cual se almacenará el valor Crisp en cuestión.
- **LABEL:** Igualmente, un valor de tipo de etiqueta sólo necesita un parámetro para almacenar el identificador asociado a dicha etiqueta (FUZZY_ID). Ese indicador es útil para poder acceder a la FMB y obtener la descripción asociada a la etiqueta.
- **INTERVALO:** Necesita los dos valores extremos del intervalo $[n, m]$, que son almacenados en F1 y F4 respectivamente.
- **APROXIMADAMENTE:** Este valor sólo necesita un valor que se almacena en F1 y que es el valor central de la distribución de posibilidad triangular. Sin embargo, para reducir operaciones (tanto matemáticas como de acceso a datos), se aprovechan los atributos F2, F3, y F4 para almacenar los valores d-margen, d+margen y margen, respectivamente. El valor margen es un valor almacenado en la FMB para cada atributo difuso, y su valor depende del significado de dicho atributo.
Esto nos permite la posibilidad de almacenar valores aproximado sin indicar el margen, usando el margen por defecto almacenado en la FMB, o también almacenar valores aproximados con un margen particular para ellos, distintos al de la FMB.
- **TRAPECIO:** Necesita forzosamente almacenar los cuatro valores que identifican a un trapecio $[\alpha, \beta, \gamma, \delta]$. En F2 y F3 se almacenan unas operaciones que simplifican las ecuaciones cuando se opera con este tipo de datos.

En esta presentación se ha primado los aspectos siguientes según Medina (1994):

Velocidad de ejecución frente a economía de almacenamiento. Para algunos de los tipos que pueden recoger este atributo, se podría emplear una representación más compacta, sin embargo, esto garantizaría la mayoría de las operaciones implicadas en una consulta (en realidad bastaría con los atributos FT y F1, almacenando los valores de cada trapecio en una tabla independiente, asociada a cada tabla que contenga atributos difusos Tipo 2).

Uniformidad en la presentación. Empleamos cinco atributos clásicos para representar los atributos difusos de este tipo.

Uso de los elementos del SGBDR anfitrión para representar la información respetando en dicha representación el esquema relacional. Este criterio, posibilitará el reducir cualquier operación de naturaleza imprecisa a términos del modelo relacional clásico.

Tipo de valores	Atributos de la BD para cada Tipo 2				
	FT	F1	F2	F3	F4
UNKNOWN	0	NULL	NULL	NULL	NULL
UNDEFINED	1	NULL	NULL	NULL	NULL
NULL	2	NULL	NULL	NULL	NULL
CRISP	3	d	NULL	NULL	NULL
LABEL	4	FUZZY_ID	NULL	NULL	NULL
INTERVALO (n,m)	5	n	NULL	NULL	m
APROXIMADAMENTE (d)	6	d	d-margen	d+margen	margen
TRAPECIO	7	α	$\beta-\alpha$	$\gamma-\delta$	δ

Tabla IV.1: Representación interna de los atributos difusos Tipo 2.

Para un atributo difuso Tipo 2, la representación usa 5 atributos para almacenar el código del tipo que le corresponda a cada valor y los atributos F1, F2, F3 y F4 para almacenar los parámetros de cada dato. Los valores NULL que aparecen en los atributos tienen el significado de valor “no-aplicable” en el sistema SGBDR anfitrión.

Atributos difusos Tipo 3: Son atributos que recogen datos escalares (SIMPLE) o distribuciones de posibilidad (DISTRIBUCION de POSIBILIDAD) sobre dominios escalares. También aceptan datos de tipo UNKNOWN, UNDEFINED Y NULL.

Igual que en los atributos difusos Tipo 2, para atributos de este tipo tendremos que almacenar en la base de datos el tipo del valor almacenado y los datos de este valor. La FMB contabilizará cada atributo de este tipo que aparezca en la base de datos. También almacenará las “*relaciones de semejanza*” definidas sobre el dominio subyacente.

En la Tabla IV.2, mostramos el sistema utilizado para representar a los atributos difusos Tipo 3. Así, vemos que un atributo difuso Tipo 3, llamado por ejemplo F, está compuesto, de hecho, por un número variable de atributos clásicos:

FT: El *tipo de valor* que corresponde al dato que queremos almacenar. Este puede ser: UNKNOWN (0), UNDEFINED (1), NULL (2), SIMPLE (3), DISTRIBUCION de POSIBILIDAD (4).

Para este atributo se consideran lista de n parejas, con $n \geq 1$, del tipo (*valor de posibilidad, etiqueta*), (FP1, F1) ... (FPn, Fn): En estos atributos se almacenan los datos de la distribución de posibilidad que deseamos almacenar. En un valor de tipo SIMPLE sólo se usa la primera pareja y el *valor de posibilidad* debería ser 1 (para estar normalizada).

En un valor de tipo DISTRIBUCION de POSIBILIDAD se podrán almacenar hasta n parejas, donde en cada una de ellas el “*valor de posibilidad*” estará en el intervalo $[0,1]$. Se puede usar menos de n parejas dejando el resto de campos a NULL. En la FMB se almacenan las *etiquetas*, su relación de semejanza y el valor de n .

Tipos de valores	Atributos de la BD para cada Tipo 3					
	FT	FP1	F1	...	FP _n	F _n
UNKNOWN	0	NULL	NULL	...	NULL	NULL
UNDEFINED	1	NULL	NULL	...	NULL	NULL
NULL	2	NULL	NULL	...	NULL	NULL
SIMPLE	3	p	d	...	NULL	NULL
DISTRIBUCION POSIBILIDAD	4	p ₁	d ₁	...	p _n	d _n

Tabla IV.2: Representación interna de atributos difusos Tipo 3.

El valor n es el máximo número de pares (grado de posibilidad, valor) que puede representar el atributo instanciado. Este valor, que tiene que ser establecido en el momento en que se declara el atributo, acota la capacidad de ese atributo para representar distribución de posibilidad y está almacenado en la FMB.

IV.2 FMB (Fuzzy Metaknowledge Base, Base de Metaconocimiento Difuso): Definición de Tablas

Como hemos visto en el apartado anterior, existe cierto tipo de información sobre los atributos descritos, que precisa ser almacenada de una forma accesible por el sistema. La Base de Metaconocimiento Difuso, va a ser la encargada de organizar toda aquella información relacionada con la naturaleza imprecisa de estos atributos. En FIRST se contempla la Base de Metaconocimiento Difuso como una extensión del catálogo del sistema, por ello, organiza la información mediante el uso de tablas o relaciones. Los elementos del tratamiento impreciso que se almacena en la FMB son los siguientes:

Atributos de la base de datos que reciben tratamiento impreciso.

Clase de información imprecisa que recogen:

- De qué Tipo difuso son estos atributos (Tipo 1, 2 ó 3) y la longitud máxima de las distribuciones de posibilidad para los atributos Tipo 3.
- Objetos definidos en el ámbito de la base de datos, como por ejemplo cuantificadores difusos de consultas.

Objetos difusos sobre cada atributo:

- Etiquetas lingüísticas (para atributos difusos Tipo 1, 2 y 3).
- El margen para valores apropiados y la distancia mínima para que dos valores sean considerados como “*muy*” separados (para atributos Tipo 1 y 2).
- Relaciones de semejanza (para atributos difusos Tipo 3).
- La descripción de esos objetos.

A continuación detallamos como FIRST implementa la FMB, que es el núcleo básico de FIRST. La organización de las tablas que la constituyen y la relación entre ellas se muestran en la Figura IV.1 y Tabla IV.3. También se han creado unas vistas que facilitan el acceso a determinados conjunto de datos.

En una base de datos relacional cada atributo es asignado unívocamente a una pareja de datos (OBJ#, COL#), donde OBJ# es el indicador de una tabla y COL# es el identificador de la columna a atributo concreto dentro de esa tabla. Nosotros en lo sucesivo nos referiremos a OBJ# como el identificador de una tabla, aunque este también puede ser el identificador de una vista. Así, si tenemos una o varias tablas difusas, podremos definir una vista difusa sobre ellas.

Para cada tabla/vista de la FMB se ha creado un sinónimo público con las siglas de la tabla, de forma que se facilita el acceso, ya que los nombres de estas tablas son largos de escribir. En la Tabla IV.3 se expone el conjunto de las tablas y vistas de la FMB y el nombre del sinónimo público creado para cada una. Además, los permisos de acceso para cada tabla han sido sobre estos sinónimos.

Tabla/ vista	Sinónimo
T. FUZZY_COL_LIST	FCL
T. FUZZY_OBJECT_LIST	FOL
T. FUZZY_LABEL_DEF	FLD
T. FUZZY_APPROX_MUCH	FAM
T. FUZZY_NEARNESS_DEF	FND
T. FUZZY_COMPATIBLE_COL	FCC
T. FUZZY_QUALIFIERS_DEF	FQD
V. FUZZY_FOR_OBJCOL	LFOC
V. FUZZY_OBJCOL_T3	LOCT3
V. ALL_COMPATIBLES_T3	ACT3

Tabla IV.3: Tablas y vistas de FIRST y sus sinónimos.

A continuación se describen cómo se crea cada tabla en la FIRST, su estructura su significado y el de cada uno de sus atributos.

Tabla FUZZY_COL_LIST

Esta Tabla contienen una descripción de aquellos atributos de la base de datos que son susceptibles de tratamiento difuso. La tabla se crea con la siguiente sentencia:

Tabla FUZZY_COL_LIST

Esta tabla contiene una descripción de aquellos atributos de la base de datos que son susceptibles de tratamiento difuso. Los atributos de esta tabla tienen el siguiente significado:

- OBJ#: Almacena el número de objeto de la tabla que tiene un atributo difuso.
- COL#: Almacena el número de columna dentro de la tabla que admitirá un tratamiento difuso.
- F_TYPE: Almacena el tipo de atributo difuso de la columna identificada por (OBJ#,COL#). Este tipo puede tomar el valor 1, 2 ó 3.
- LEN: Almacena la longitud máxima de una distribución de posibilidad en atributos Tipo 3 , el número máximo de parejas (valor de posibilidad, etiqueta) que admite una distribución de posibilidad en este atributo.
- COM: Almacena un comentario opcional.

Tabla FUZZY_OBJECT_LIST

Esta tabla contiene una lista de los objetos de tipo difuso que hay definidos en las columnas de la base de datos. Los atributos de esta tabla tienen el siguiente significado:

- (OBJ#,COL#): Almacena el identificador del atributo al que aparece el objeto.
- FUZZY_ID: Identificador del objeto difuso.
- FUZZY_NAME: Nombre del objeto sin espacios.
- FUZZY_TAPE: Tipo del objeto. Puede ser uno de los siguientes:
 - 1 para escalares sujetos a tratamiento mediante una relación de semejanza definida en FUZZY_NEARNESS_DEF.

- 2 para cualificadores definidos sobre el índice de cumplimiento en la consulta (umbral) y almacenados en la tabla FUZZY_QUALIFIERS_DEF.
- 3 para etiquetas lingüísticas definidas sobre cuantificadores relativos y almacenadas en la tabla FUZZY_LABEL_DEF con $\alpha, \beta, \gamma, \delta \in [0,1]$.
- 4 para etiquetas lingüísticas definidas sobre cuantificadores absolutos y almacenados en la tabla FUZZY_LABEL_DEF con $\alpha, \beta, \gamma, \delta \geq 0$.

Tabla FUZZY_LABEL_DEF

Esta tabla contiene los puntos que determinan la distribución de posibilidad trapezoidal correspondientes a los tipos de objetos 0, 3 y 4 del atributo FUZZY_TYPE de la tabla FUZZY_OBJECT_LIST. Los campos de esta tabla tienen el siguiente significado:

- (OBJ#,COL#, FUZZY_ID): Estos tres campos son la llave primaria de esta tabla y llave externa a la tabla FUZZY_OBJECT_LIST.
- ALFA, BETA, GAMMA Y DELTA: Definen una distribución de posibilidad trapezoidal.

Tabla FUZZY_APROX_MUCH

Esta Tabla almacena datos que son utilizados cuando se trabaja con atributos difusos Tipo 1 ó 2. Sus campos tienen el siguiente significado.

- (OBJ#,COL#): Almacena el identificador del atributo al que aparece el objeto.
- MARGEN: Es el margen utilizado en las etiquetas triangulares de valores aproximados.
- MUCH: Es el valor que indica la distancia mínima que 2 valores de este atributo sean considerados como muy separados.

Nota: En esta tabla está la restricción que obliga a que el valor MARGEN sea menor que el de MUCH.

Tabla FUZZY_NEARNESS_DEF

Esta tabla presenta las medidas de proximidad, semejanza o similitud entre los diferentes valores de dominio permitidos sobre los campos de Tipo 3. Sus atributos tienen el siguiente significado:

- (OBJ#,COL#): Almacena el identificador del atributo Tipo 3 que posee la función de semejanza.
- FUZZY_ID1: Identificador de un objeto etiqueta. Los atributos (OBJ#,COL#,FUZZY_ID1) forman una llave externa que debe existir en la tabla FUZZY_OBJECT_LIST y en esta tabla su campo FUZZY_TYPE correspondiente debe tener el valor 1.
- FUZZY_ID2: Identificador de otro objeto etiqueta. Los atributos (OBJ#,COL#,FUZZY_ID2) forman una llave externa que debe existir en la tabla FUZZY_OBJECT_LIST y en esta tabla su campo FUZZY_TYPE correspondiente debe tener el valor 1.
- DEGREE: Grado de similitud o semejanza entre las etiquetas indicadas por los 2 atributos anteriores (FUZZY_ID1y FUZZY_ID2). Este valor estará comprendido en el intervalo [0,1] y se admite un máximo de 2 números decimales.

Tabla FUZZY_COMPATIBLE_COL

Indica los atributos difusos Tipo 3 que son compatibles con otros. De esta forma no es necesario definir las etiquetas y las relaciones de similitud para cada uno de ellos. Sus campos tienen el siguiente significado:

- (OBJ#1,COL#1):Almacena el identificador del atributo Tipo 3 que es compatible con otro. Indica que este atributo no tiene etiquetas definidas sobre él y tomará las etiquetas por el siguiente atributo.
- (OBJ#2,COL#2): Almacena el identificador de un atributo Tipo 3 que tiene etiquetas y relaciones de semejanza definidas sobre él y que serán adoptadas por el atributo anterior.

Tabla FUZZY_QUALIFIERS_DEF

La utilidad de ésta tabla es mínima, por lo que no está implementada en el servidor FSQ, por lo cual no es relevante incluir sus características.

La Figura IV.1 muestra el esquema de la FIRST, en la cual se almacenan atributos difusos Tipo 2 y Tipo 3, además de cuantificadores difusos.

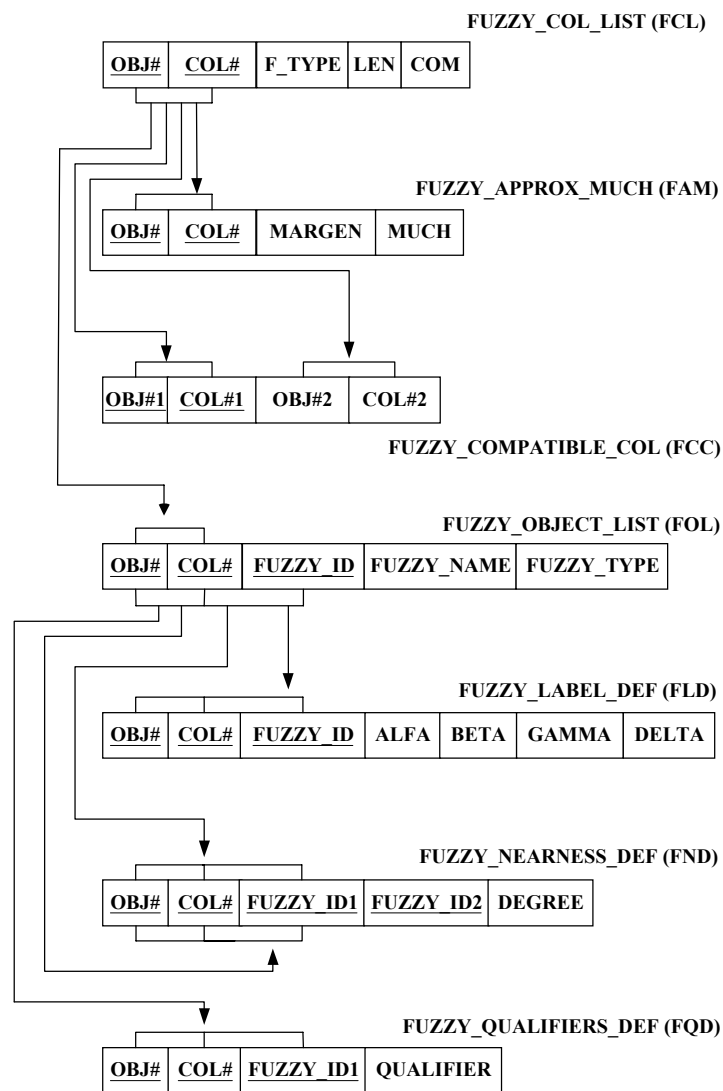


Figura IV.1: Esquema de la FIRST de Medina (1994).

IV.3 Implementación de la FIRST: Caso del Control de la Calidad del Papel

Una vez realizado el modelo conceptual mostrado en la Figura 5.24, y detallado sus atributos y restricciones en el apartado 5.2, a continuación mostramos la implementación de los datos difusos, para ser incorporados al FSQL de Galindo (1999), utilizando la FIRST propuesta en Medina (1994). La implementación ha sido en una base de datos relacional Oracle 8, también se consideran la transformación de ER/EER a MR (Luque et al., 2001)).

Tabla Usos: La Tabla Usos se forma a partir del tipo de entidad que lleva el mismo nombre, y de este tipo de entidad toma los atributos Cod_Usos y T3:Usos. Del conjunto de atributos de la tabla, la clave principal es el atributo Cod_Usos. Ninguno de los atributos de la tabla mantiene referencia alguna con atributos de otras tablas. La tabla Usos queda de la forma:

Usos (Cod_Usos, T3:Usos)

Como el atributo difuso Usos es Tipo 3, su representación en la FIRST es la siguiente:

Cod_Uso	UsosT	UsosP1	Usos1	...	UsosP5	Usos5



Atributo Difuso Usos Tipo 3 según Tabla IV.2

Figura IV.2: Representación de la relación Usos.

Tabla Gramaje: La Tabla Gramaje se forma a partir del tipo de entidad que lleva el mismo nombre, y de este tipo de entidad toma los atributos Cod_Gramaje y T1:Gramaje. Del conjunto de atributos de la tabla, la clave principal es el atributo Cod_gramaje. Ninguno de los atributos de la tabla mantiene referencia alguna con atributos de otras tablas. Como esta Tabla contiene atributos Tipo 1 su representación es la misma. La tabla Gramaje queda de la forma:

Gramaje (Cod_Gramaje, T1:Gramaje)

Tabla Cartulinas Estucadas: La Tabla Cartulinas Estucadas se forma a partir del tipo de entidad que lleva el mismo nombre, y de este tipo de entidad toma los atributos Cod_Cartulina, Cod_capa Tono_Cara, Tono_Reverso e Impresión. Del conjunto de atributos de la tabla, la clave principal es el atributo Cod_Cartulina. El atributo Cod_capa de la tabla mantiene referencia con la Tabla Capas. La tabla Cartulinas Estucadas queda de la forma:

Cartulinas Estucadas (Cod_Cartulina, Cod_capa, T3:Tono_Cara, T3:Tono_Reverso, Impresión)

Como los atributos difusos de ésta relación son de Tipo 3, su representación en la FIRST es la siguiente:

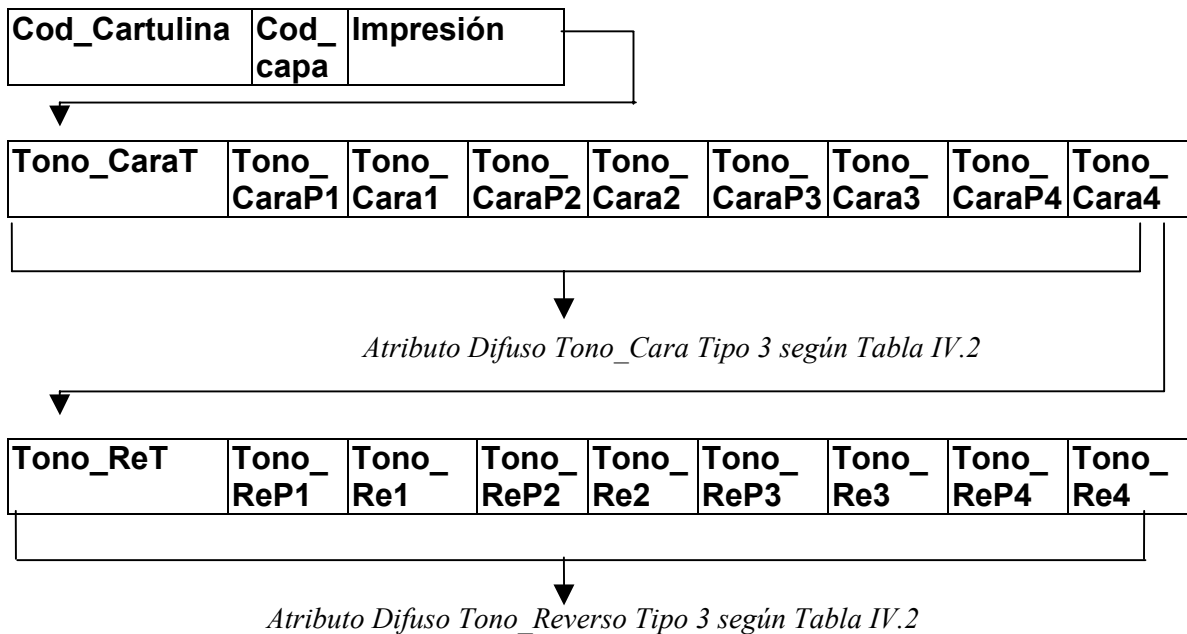


Figura IV.3: Representación de la relación Cartulinas Estucadas.

Considérese que en los atributos Tono_Cara y Tono_Reverso se almacenarán listas de cuatro parejas (valor de posibilidad, valor) ya que los datos que deseamos almacenar son sólo cuatro colores. Véase Tabla 5.4.

Tabla Capas: La Tabla Capas se forma a partir del tipo de entidad que lleva el mismo nombre, y de este tipo de entidad toma los atributos Cod_Capa y T3:Capas. Del conjunto de atributos de

la tabla, la clave principal es el atributo Cod_Capas. Ninguno de los atributos de la tabla mantiene referencia alguna con atributos de otras tablas. La tabla Capas queda de la forma:

Capas (Cod_Capas, T3:Capas)

El atributo difuso Tipo de capas es Tipo 3, su representación en la FIRST es la siguiente:



Figura IV.4: Representación de la relación Capas.

Considérese que el atributo Tipo de Capa se almacenarán la lista de 7 parejas generadas por las subclases (valor de posibilidad, valor). Véase Figura 5.22.

Tabla Pilas: La Tabla Pilas se forma a partir del tipo de entidad que lleva el mismo nombre, y se deriva de una especialización del tipo de entidad Cartulinas Estucadas y por tanto mantiene una referencia a través del atributo Cod_Cartulina con la tabla Cartulinas Estucadas, además de incorporar los atributos Cod_Pila, Formato_largo, Formato_ancho, Peso, Estado. Del conjunto de atributos de la tabla, la clave principal es la agregación de los atributos Cod_Pilas y Cod_Cartulina.. La tabla Pilas queda de la forma:

Pilas (Cod_Pilas, Cod_Cartulina, T2:Formato_Largo, T2:Formato_ancho, T2:Peso, T3:Estado)

Como los atributos difusos de ésta relación son de Tipo 2 y 3, su representación en la FIRST es la siguiente:

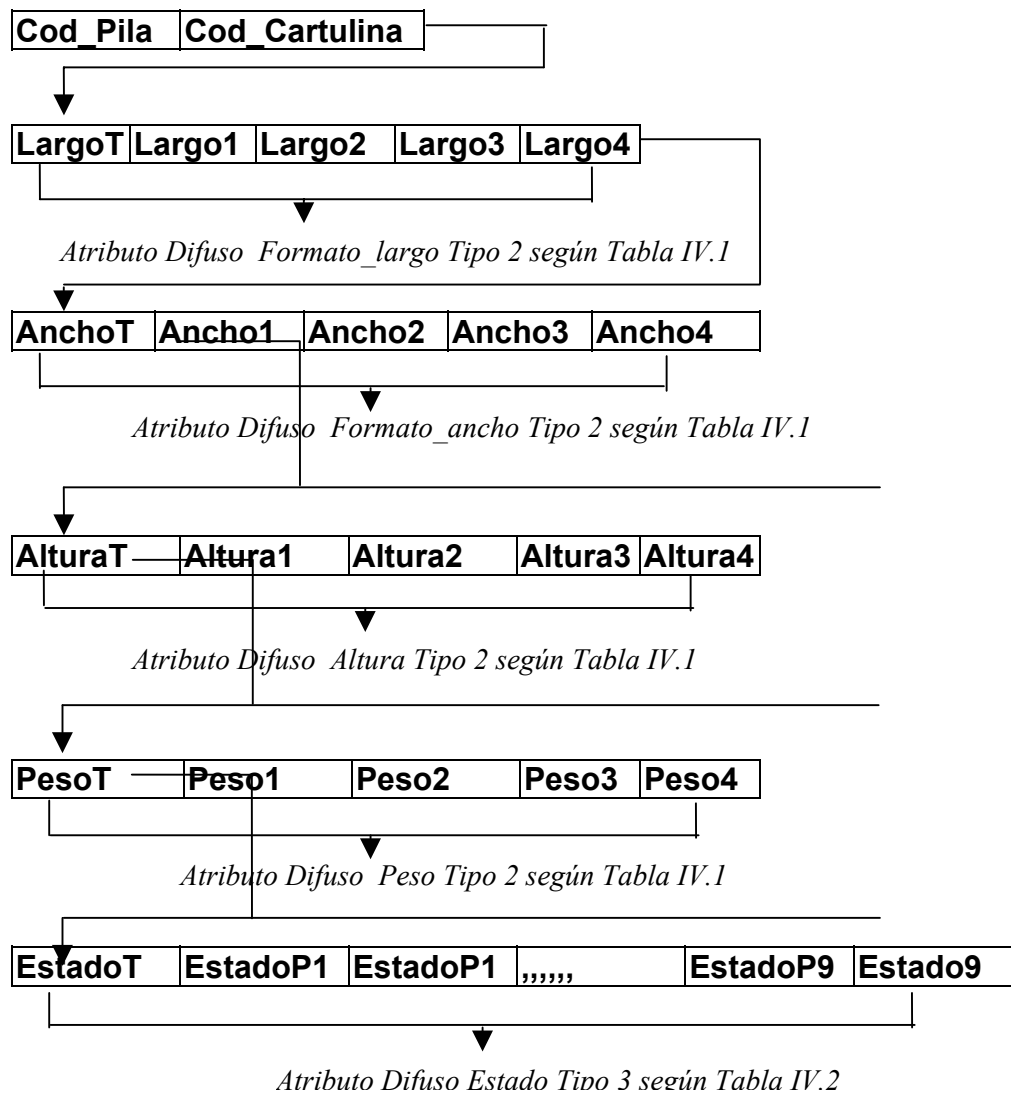


Figura IV.4: Representación de la relación Pilas.

Tabla Rollos: La Tabla Rollos se forma a partir del tipo de entidad que lleva el mismo nombre, y se deriva de una especialización del tipo de entidad Cartulinas Estucadas y por tanto mantiene una referencia a través del atributo Cod_Cartulina con la tabla Cartulinas Estucadas, además de incorporar los atributos Cod_Rollo, Formato_largo, Formato_ancho, Peso, Estado y Tipo. Del conjunto de atributos de la tabla, la clave principal es la agregación de los atributos Cod_Rollos y Cod_Cartulina. La tabla Rollos queda de la forma:

Rollos (Cod_Rollos, T2:Formato_largo, T2:Formato_ancho, T2:Peso, T3:Estado, Tipo)

Como los atributos difusos de ésta relación son de Tipo 2 y 3, su representación en la FIRST es la siguiente:

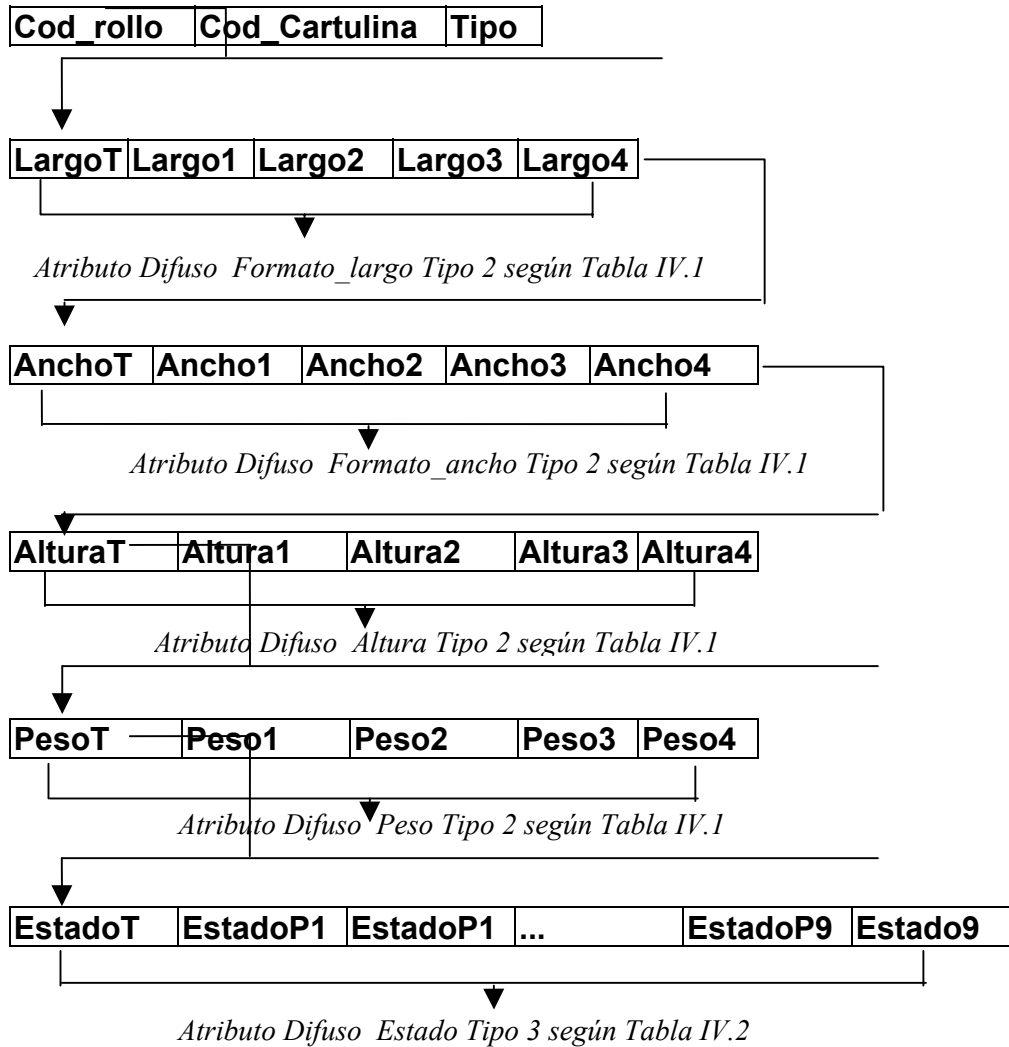


Figura IV.5: Representación de la relación Rollos.

Tabla Tipo_Cart: La Tabla Tipo_Cart se forma a partir del tipo de atributo Tipo 3 Tipo_Cartulina, esta tabla incorpora los atributos Cod_Tipo y T3:Nombre. Ninguno de estos atributos mantiene referencia con atributos de otras tablas, sino que es referenciada por la tabla Gr_Tipo_Cart_Usos. El identificador de esta tabla es el atributo Cod_Tipo. Esta tabla queda de la forma:

Tipo_Cart (Cod_Tipo, T3:Nombre)

Como la relación tiene un atributo difuso Tipo 3, su representación en la FIRST es la siguiente:

Cod_Tipo	NombreT	NombreP1	Nombre1	...	NombreP4	Nombre4

Atributo Difuso Nombre Tipo 3 según Tabla IV.2

Figura IV.6: Representación de la relación Tipo_Cart.

Tabla Gr_Tipo_Cart_Usos: La Tabla Gr_Tipo_Cart_Usos se forma a partir de tipo de interrelación *Tiene* que existe entre los tipos de entidad Cartulinas Estucadas, Usos y Gramajes. Esta Tabla tiene los atributos Cod_Cartulina, Cod_Usos, Cod_Gramaje y Cod_Tipo en los cuales:

El atributo *Cod_Carulina* es derivado del tipo de entidad Cartulinas Estucadas por ser el identificador de éste, y a través de este atributo mantiene referencia con la Tabla Gr_Tipo_Cart_Usos.

El atributo *Cod_Usos* es derivado del tipo de entidad Usos por ser el identificador de éste, y a través de este atributo mantiene referencia con la Tabla Gr_Tipo_Cart_Usos.

El atributo *Cod_Gramaje* es derivado del tipo de entidad Gramaje por ser el identificador de éste, y a través de este atributo mantiene referencia con la Tabla Gr_Tipo_Cart_Usos.

Además, es agregado el atributo *Cod_Tipo* el cual es derivado del tipo de entidad Tipo_Cart por ser el identificador de éste, y a través de este atributo mantiene referencia con la Tabla Gr_Tipo_Cart_Usos.

La clave principal de la Tabla Gr_Tipo_Cart_Usos es la agregación de los atributos Cod_Cartulina, Cod_Usos, Cod_Gramaje y Cod_Tipo, la cual queda de la forma:

Gr_Tipo_Cart_Usos (Cod_Cartulina, Cod_Usos, Cod_Gramaje, Cod_Tipo)

En este caso todos los atributos son clásicos, por lo que su representación queda igual.

Concluyendo, con lo expuesto en este apartado y en el apartado 5.2 se puede implementar gran parte del modelo de datos propuestos utilizando el modelo relacional extendido para bases de datos difusas con el servidor FSQL.

Apéndice V: Museos Digitales en Internet: Modelo EER Difuso y Recuperación de Imágenes Basada en su Contenido

V.1 EER Difuso de la Base de Datos Museos Digitales

El modelo de datos propuesto en la Figura V.1 está simplificado por motivos de espacio, evitando poner los atributos de cada entidad y otras entidades que también serían interesantes (como para la gestión de socios o personas registradas, venta de entradas, guías del museo, etc.). En síntesis, se han identificado las entidades y las relaciones existentes entre ellas, así como los requisitos deseables del sistema.

En el modelo conceptual se pueden distinguir algunas entidades tales como: *Museo* que alberga distintos *Objetos*, los cuales pertenecen a ciertos *Autores*. Además, cada objeto podrá publicar una o varias *Imágenes*. Los atributos para la entidad *Museo* son tales como Director (que puede ser otra entidad), Dirección, Fecha de inauguración, etc.. Como atributos difusos (Tipo 2) podemos identificar el número de objetos expuestos y no expuestos (de los fondos del museo). También puede incluirse un atributo Calidad que evalúe (de 0 a 10) el nivel del museo en general. Los atributos de *Autor* son tales como Nombre, Lugar de nacimiento, Biografía (más o menos breve). Como atributos difusos pueden incluirse las Fechas de nacimiento y defunción, entendiendo que estas fechas incluyen (día, mes, año y siglo). Obsérvese que esto permite preguntar por autores que nacieran “aproximadamente” en cierta fecha o siglo.

Para la entidad *Objeto* tenemos los atributos no difusos Nombre (o Título), una descripción textual y una breve Historia del mismo (si procede). Como atributos difusos (Tipo 2) podemos establecer la Fecha de realización y las medidas: Alto, Ancho, Profundo y Peso (estos últimos también podrían considerarse como Tipo 1, impidiendo así la posibilidad de almacenar información no exacta). En esta entidad hay unos atributos difusos Tipo 3 muy interesantes. En efecto, el atributo Estilo almacenaría el estilo de la obra teniendo en cuenta que hay obras que pueden pertenecer a diversos estilos y que buscar por cierto estilo puede implicar interés en recuperar obras de estilos similares (en caso de que no se encuentre lo deseado o se encuentren pocas obras). Un ejemplo de este atributo fue puesto anteriormente. El atributo Material permitiría guardar información sobre el material o materiales utilizados (Mármol, Granito, Óleo sobre lienzo, Óleo sobre tabla, Acuarela, etc.). Así, buscar una obra en Mármol puede recuperar obras en otros materiales similares si el usuario así lo desea. Otro atributo interesante es el de

Tema, cuyos valores identifican brevemente el asunto que representa (Histórico, Mitológico, Paisaje, Animal, etc.).

Un atributo Tipo 3 importante es el Tipo de Objeto, el cual es utilizado para clasificar los objetos en distintas subclases (ver Figura V.1), las cuales pueden tener sus propios atributos (si es una Pintura interesará si la escuela es la española, flamenca, alemana, etc.).

Con respecto a la entidad *Imagen* podemos almacenar distintos atributos que serían de dos tipos: textuales y calculados. El atributo Descripción será de tipo textual y describirá la imagen (evitando hacerlo si coincide con la descripción del objeto). Los atributos calculados son aquellos cuya información se extrae de forma automática a partir de la imagen.

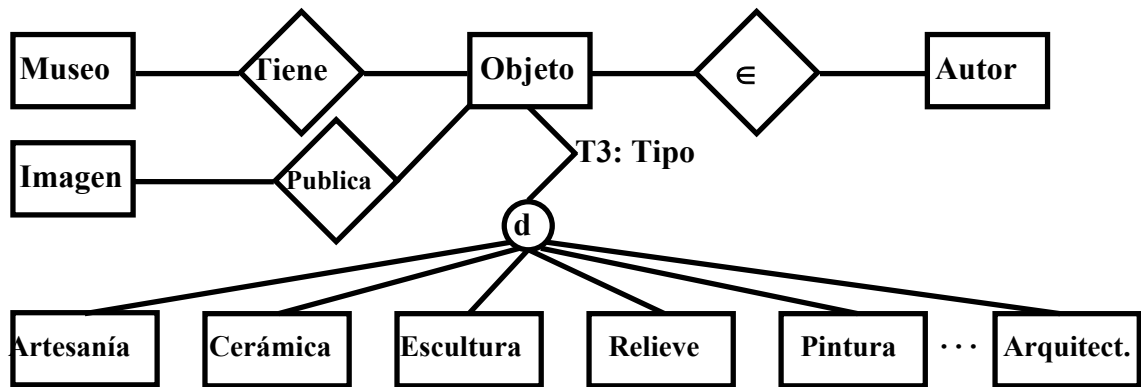


Figura V.1. Modelo EER difuso simplificado, sin atributos, de museos digitales.

V.2 Recuperación de Imágenes Basada en su Contenido

Los últimos años han visto un rápido incremento del tamaño de las colecciones de imágenes digitales (médicas, históricas, policiales, pictóricas, etc.). Esta situación plantea una necesidad básica: la búsqueda y obtención de las imágenes deseadas entre una extensa colección. Los métodos tradicionales de clasificación de la información (anotaciones textuales, establecimiento de taxonomías) se muestran insuficientes e inadecuados por diversos motivos. Básicamente podemos destacar dos problemas: (i) la cantidad de trabajo que requiere asociar un texto a cada imagen y (ii) expresar el rico contenido de una imagen en texto no es fácil y además es subjetivo. Además la subjetividad e imprecisión de las anotaciones pueden causar errores en la recuperación.

Para solucionar esos problemas surgió, a principios de los '90, la recuperación de imágenes basada en el contenido (*content-based image retrieval*). En ella las imágenes son indexadas y recuperadas por su propio contenido visual, como el color o la textura. La recuperación de imágenes basada en el contenido se fundamenta en tres aspectos: la extracción de características visuales, la indexación multidimensional (para facilitar una búsqueda rápida) y el diseño del sistema de recuperación. Las características visuales se clasifican en generales y específicas. Estas últimas dependen de la aplicación. Las características visuales generales que son usadas en la mayoría de las aplicaciones son:

- **Color:** Es una de las características más usadas. Es relativamente robusta a las variaciones del fondo e independiente del tamaño y orientación de la imagen. Las técnicas más usadas como representación del color son el histograma de color, el histograma de color acumulado, los momentos de color y los conjuntos de color.
- **Textura:** Con esto nos referimos a patrones visuales homogéneos formados por diversos colores o intensidades. Es una propiedad innata de prácticamente todas las superficies, como nubes, árboles, pelo o ladrillos. Las características de textura se suelen representar usando una matriz de concurrencia, propiedades psicológicas (contraste, regularidad, tosquedad, aspereza, etc.), transformadas wavelet.
- **Formas de objetos en la imagen:** Algunas aplicaciones requieren que la representación de la forma sea invariante a traslación, rotación y escalado mientras que otras no. En general, las representaciones de la forma se dividen en dos categorías, las basadas en contornos y las basadas en regiones. Las primeras usan sólo el contorno exterior de la forma mientras que las últimas usan la región de la forma completa. Se han desarrollado numerosos métodos para ambas categorías, pero los más representativos son los descriptores de Fourier (transformada de Fourier del contorno) para la primera, y los momentos invariantes (momentos basados en regiones que sean invariantes a transformaciones) para la segunda.
- **Diseño del color:** Se trata de usar conjuntamente la característica de color y las relaciones espaciales. Una aproximación sencilla es dividir la imagen en bloques y extraer las características de color de cada bloque. Otra aproximación es segmentar la imagen en regiones con características de color destacadas y luego almacenar el conjunto de características de color y la posición de cada región. Su desventaja es la problemática que supone la segmentación de una imagen. Otras técnicas son usar momentos de color sobre regiones, usar una matriz de concurrencia por ejemplo, de color.

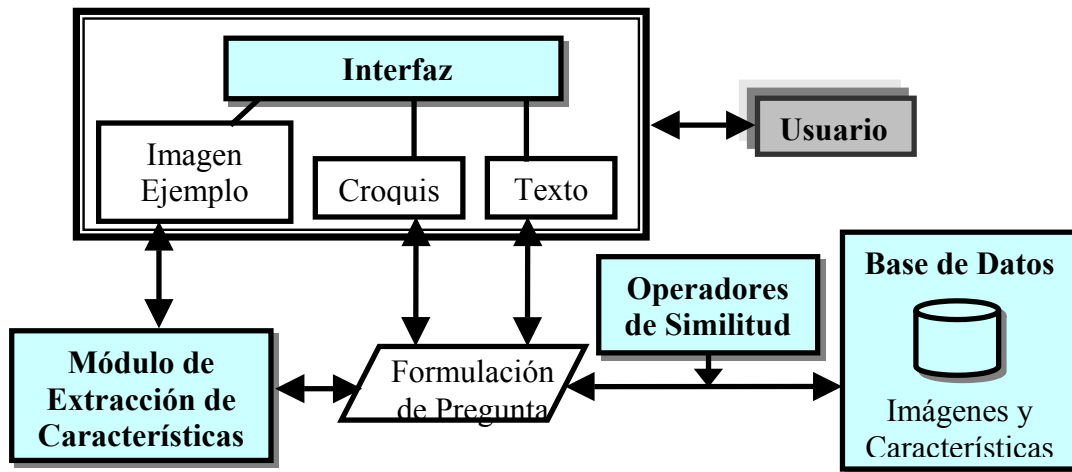


Figura V.2: Arquitectura de un sistema de recuperación de imágenes sobre una BD.

Existen numerosos sistemas de recuperación de imágenes basados en el contenido, los cuales incluyen alguna o varias de las características anteriores para hacer la búsqueda. La mayoría son sistemas de propósito general y faltan estudios en los que éstos se usen en aplicaciones prácticas para investigar las ventajas e inconvenientes de las distintas opciones. Algunos de los más interesantes son: QBIC (<http://www.qbic.almaden.ibm.com>), MARS (<http://jadzia.ifp.uiuc.edu:8000>).

V.3 Sistema de Recuperación de Imágenes MUSEUM

El diseño general del sistema de recuperación de imágenes que proponemos y al que hemos denominado MUSEUM se muestra en la Figura V.2. Destacamos tres componentes principales, además de la base de datos explicada anteriormente:

- **Interfaz:** Va a permitir tanto la confección de la pregunta como la presentación del resultado de la misma. La pregunta puede ser en base a (i) una imagen ejemplo, (ii) un dibujo o croquis contruidos por el usuario, o bien (iii) por texto (incluyendo palabras clave o notas).
- **Módulo de extracción de características:** Es el elemento más importante de todo el proceso. Debe abstraer la información considerada relevante para quedar incluida en la base de datos (en los atributos de cada imagen individualmente).
- **Función u operadores de similitud:** Permiten indexar y recuperar la información a partir de la pregunta realizada.

V3.1 Módulo de Extracción de Características

A cada imagen se le asocia un vector de características (atributos), que incluye información sobre color, relaciones de color y formas, y que se detalla a continuación.

Color y Relaciones de Color: Una imagen en color se representa en un espacio de color tridimensional. Existen numerosos espacios de representación del color. El más conocido es el RGB (*red-green-blue*), que es además el que utilizamos. En este espacio, cualquier color se representa como un vector tridimensional indicando la proporción de rojo, verde y azul. Un primer atributo de color que utilizaremos es el *histograma de color entre otros*. Estadísticamente, un histograma representa la probabilidad de que un píxel de la imagen tome un determinado color (tríada de colores). Si suponemos que u es una variable aleatoria que representa un nivel de color en una imagen, el histograma se representa por la ecuación (1).

$$p_u(x) = \text{prob}[u = x] \cong \frac{\text{Número de píxeles con color } x}{\text{Número total de píxeles en la imagen}} \quad (1)$$

donde $x=(r_i, g_j, b_k)$, $i, j, k=0, \dots, L-1$, siendo L el número de niveles de color. Fácilmente se deduce que este histograma puede ser muy grande si el número de niveles de cada componente de color es elevado. Por ejemplo, usando una representación RGB, si se tienen 256 niveles de cada componente, el histograma tendría $256 \times 256 \times 256$ combinaciones posibles de colores. Este problema se puede solucionar cuantizando el histograma, es decir, reduciendo el número de niveles de color.

También utilizamos los *momentos* m_i y los *momentos centrales* μ_i sobre el histograma, que

$$m_i = \sum_{x=0}^{L-1} x^i p_u(x), \quad i=1, 2, \dots \quad (2)$$

se definen como muestran las siguientes ecuaciones (Sonka, 1998):

Como la mayoría de la información de la distribución de color se concentra en los momentos de orden bajo, utilizamos sólo el primer momento m_1 (media) y el segundo y tercer

$$\mu_i = \sum_{x=0}^{L-1} (x - m_1)^i p_u(x), \quad i=1, 2, \dots \quad (3)$$

momentos centrales μ_2 y μ_3 (varianza y desviación o sesgo respectivamente).

Para representar las relaciones de color dividimos la imagen en 4 regiones, superior izquierda, superior derecha, inferior izquierda e inferior derecha, las cuales se solapan entre sí para aumentar la estabilidad de los cálculos. Para cada una de ellas calculamos el histograma de color y conservamos los momentos m_1 , μ_2 y μ_3 . Así reducimos la cantidad de información para almacenar y comparar.

En definitiva, guardamos para cada imagen los siguientes **atributos de color**: histograma cuantizado, momentos sobre el histograma de la imagen completa y momentos sobre el histograma de cada una de las 4 regiones.

Formas: Extraer las formas más significativas de una imagen es algo complejo. Se complica aún más si los objetos están superpuestos, como lo están en imágenes reales. En nuestro caso será más simple en fotos de esculturas u objetos de cerámica en las que una única figura destaca sobre el fondo y más complejo en fotos de cuadros. El primer paso importante es el de aislar las figuras que se pretenden caracterizar. El proceso de segmentación para conseguir subdividir la imagen en sus partes constituyentes es una de las tareas más difíciles del procesamiento de datos. Distintos algoritmos de segmentación se pueden encontrar en. Cada objeto de la imagen lo caracterizamos a través de su contorno siguiendo los siguientes pasos:

1. **Cálculo de puntos frontera:** Usamos el algoritmo de Canny.
2. **Seguimiento del contorno:** Se trata de obtener una secuencia de puntos consecutivos que no se intersecten y que definan la frontera de un objeto. Se calcula conectando los puntos frontera obtenidos en la etapa anterior.
3. **Cálculo de la curva de curvaturas:** Esta curva contiene información sobre el grado de curvatura que presenta el contorno en cada uno de los puntos del mismo.
4. **Caracterización del contorno:** Proponemos una caracterización difusa del contorno. En primer lugar, se calculan los máximos locales de la curva de curvaturas. Estos puntos corresponderán con las esquinas del objeto en cuestión, y los llamamos **puntos característicos**. El contorno se representa, por tanto, como un conjunto de puntos característicos. Para cada uno de ellos se almacena su grado de curvatura como una etiqueta lingüística (alto, medio o bajo grado de curvatura) y la distancia de ese punto al anterior (también como una etiqueta lingüística: lado largo, corto o medio). Cada característica es tratada como un atributo difuso. Así, podemos considerar un contorno que tiene “aproximadamente 5 vértices” con “alto grado de curvatura” situados “aproximadamente” a la misma distancia. Obsérvese que esta caracterización es invariante frente a rotación, escalado y traslación.

Finalmente, la representación de las formas más significativas de la imagen vendrá dada por un vector de contornos difusos. Cada contorno difuso estará representado por el número de puntos característicos, el grado de curvatura de cada uno y la distancia entre vértices, todos ellos representados como atributos difusos.

V.3.2 Módulo de Recuperación

Para realizar la búsqueda es necesario establecer una función que indique el grado de ajuste entre el contenido de la consulta y las distintas imágenes de la base de datos. La **función de similitud** definida atenderá a dos elementos de caracterización:

1. El **color** a partir de una medida de distancia entre los distintos atributos. En primer lugar, para comparar los histogramas se utiliza una métrica que tiene en cuenta similitudes entre colores parecidos pero no idénticos. Para comparar las medidas de momentos se utiliza la distancia Euclídea ponderada. Para calcular la similitud total se hace la media entre todas las medidas obtenidas.
2. La **forma** a partir de una caracterización de los objetos de interés usando el contorno difuso de los mismos. La comparación se hace utilizando un lenguaje de consultas difuso FSQL (Galindo et al., 1998) que nos va a permitir comparar de forma flexible el número de puntos característicos, la distancia entre vértices y el grado de curvatura, todos ellos tomados como atributos difusos.

El sistema de búsqueda permite, además, preguntar por las distintas características combinadas asociando un peso a cada una (entre 0 y 5). Si el peso dado es 0 esta característica no es tomada en cuenta. Se pretende, además, incorporar nuevos atributos como la textura en el sistema de búsqueda.

Apéndice VI: Acrónimos

AECI: Agencia Española de Cooperación Internacional.

CYTED: Ciencia y Tecnología para el Desarrollo.

ER: Entity-Relationship.

EER: Enhanced Entity-Relationship.

ERD: Diagrama Entidad Relación.

FIRST: Fuzzy Interface for Relational SysTems.

FOOD: Fuzzy Object Oriented Diagrams.

FuzzyEER: Fuzzy Enhanced Entity-Relationship.

FSQL: Lenguaje de Consulta Estructurado Difuso.

FUPME: Fuzzy Update Protocol Model Expressions.

FRSL: Fuzzy Requirement Specification Language.

FDNER: Fuzzy Diagrams Nested Entity Relationship.

FRDBS: Fuzzy Relational Data Base System.

FEER: Entidad Relación Extendido Difuso

FOODB: Base de Datos Orientada a Objeto Fuzzy.

GEFRED: Generalized Model of Fuzzy Relational Databases

RITOS2: Red IberoAmericana de Tecnologías del Software para la década del 2000.

SQL: Lenguaje de Consulta Estructurado.

SGBD: Sistema Gestor de Bases de Datos.

UML: Lenguaje Unificado de Modelado.