



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Comparación cuantitativa de
ontologías naturales

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

ING. ÁNGEL REFUGIO MANDUJANO GARCÍA

DIRECTORES:

DR. JESÚS GUILLERMO FIGUEROA NAZUNO

DR. FRANCISCO HIRAM CALVO CASTRO



Diciembre de 2016



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:00 horas del día 30 del mes de noviembre de 2016 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

“Comparación cuantitativa de ontologías naturales”

Presentada por el alumno:

MANDUJANO

Apellido paterno

GARCÍA

Apellido materno

ÁNGEL REFUGIO

Nombre(s)

Con registro:

B	1	4	0	4	7	7
----------	----------	----------	----------	----------	----------	----------

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

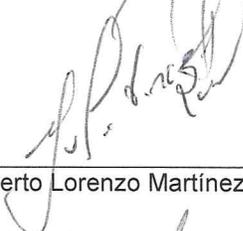
LA COMISIÓN REVISORA

Directores de Tesis

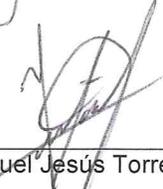

Dr. Jesús Guillermo Figueroa Nazuno


Dr. Francisco Hiram Calvo Castro

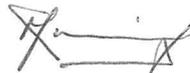

Dr. Sergio Suárez Guerra


Dr. Gilberto Lorenzo Martínez Luna


Dr. Marco Antonio Moreno Ibarra


Dr. Miguel Jesús Torres Ruiz

PRESIDENTE DEL COLEGIO DE PROFESORES



Dr. Marco Antonio Ramírez Salinas



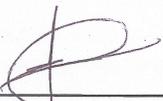


INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 07 del mes Diciembre del año 2016, el (la) que suscribe Ángel Refugio Mandujano García alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B140477, adscrito al Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Jesús Guillermo Figueroa Nazuno y Dr. Francisco Hiram Calvo Castro y cede los derechos del trabajo intitulado “Comparación cuantitativa de ontologías naturales”, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección a.mandujanogarcia@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Angel Refugio Mandujano García

Resumen

Son pocas las técnicas de alineamiento elástico que realizan una comparación global entre matrices de datos, por lo que su análisis es de gran interés en diversas áreas en ciencias de la computación, siendo de gran utilidad en reconocimiento de patrones, minería de datos y aprendizaje máquina. En el presente trabajo se describe una metodología de elicitación de conocimiento empleando Ontologías Naturales que están basadas en el uso de las Redes Semánticas Naturales; así mismo, se implementa una medida de distancia, que es una aproximación al problema NP-Completo de alineamiento elástico en matrices numéricas. La técnica de distancia matricial llamada Fréchet Moderno se ha utilizado para la comparación de conocimiento humano en diversos grupos de sujetos pertenecientes al dominio de Computación. Se incluyen en este documento diferentes análisis experimentales sobre la potencialidad de la técnica para ser aplicada en la comparación de conocimiento.

Abstract

There are few elastic matching techniques that perform a global comparison of numerical matrices, so their analysis is important in different Computer Science fields, such as Pattern Recognition, Data Mining, Machine Learning, etc. It is described in this work, a knowledge elicitation methodology that use Natural Ontologies, latter are based in the use of Natural Semantic Networks. At the same time it is implemented a distance measurement, that is an approximation to the NP-Complete problem of Elastic Matching between matrices. The matricial distance technique called Modern Fréchet had been used to compare human knowledge in different groups of people in the Computer Science domain. It is also included in this document, different empirical analysis about the distance technique potentiallity applied in the knowledge comparison.

Agradecimientos

Quiero dar mis mas sinceros agradecimientos a mis directores de tesis al Dr. Jesús G. Figueroa Nazuno por su participación, disponibilidad y paciencia durante el desarrollo de este trabajo. Gracias por las incontables pláticas interesantes llenas de “separaciones”, han sido muy enriquecedoras. Así mismo agradezco al Dr. Hiram Calvo por su entusiasmo, su apoyo y sus comentarios. Gracias a ambos por brindarme su confianza. A mis compañeros y amigos del CIC, que también se han mantenido en constante apoyo.

Agradezco especialmente al seminario del Laboratorio de Inteligencia Artificial porque es un excelente espacio para el seguimiento, la orientación y la discusión de las investigaciones. Gracias a todos los organizadores del seminario, que su labor continúe con el valioso objetivo de hacer mejores estudiantes.

¡Muchas gracias!

Se agradece al Instituto Politécnico Nacional(IPN) y al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el imprescindible apoyo brindado en la realización de este trabajo. Al mismo tiempo agradezco a la comunidad académica del Centro de Investigación en Computación quienes son el motor para el avance de la ciencia.

Dedico este trabajo a mi familia entera, que sin su apoyo incondicional y sus enseñanzas no habría sido posible llegar a la meta. Es para mi todo un honor haber aprendido muchas cosas de cada uno de ustedes.

— Ángel Refugio Mandujano García

Índice general

1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Particulares	2
1.3. Justificación	3
1.4. Contribuciones	3
1.5. Descripción de la propuesta	3
1.6. Organización del documento	4
2. Conocimiento	6
2.1. Sobre el Conocimiento	6
2.1.1. Conocimiento Humano	9
2.1.2. Modelo de memoria humana	10
2.1.3. Lenguaje	12
2.2. Elicitación de conocimiento	15
2.2.1. Red Semántica	16
2.2.2. Ontología	21
2.2.3. Redes Semánticas Naturales	28
2.2.4. Otras formas de representación basadas en memoria	35

3. Distancia	37
3.1. Definición de Similitud	39
3.1.1. Similitud y Distancia	43
3.2. Espacio Métrico	44
3.3. Distancia en R^1	46
3.4. Distancia en R^2	47
3.5. Alineación Elástica	48
3.5.1. Distancia Elástica R^1	49
3.5.2. Distancia Elástica R^2	51
4. Estado del Arte	53
4.1. Distancias basadas en Elastic Matching	53
4.2. Medidas R^2	55
4.2.1. Distancia Euclidiana	56
4.2.2. Distancia Chord	56
4.2.3. Coeficiente de Rango Spearman	57
4.2.4. Distancias Estadísticas	57
4.2.5. Comparativa de las técnicas	59
5. Fréchet Moderno	60
5.1. Distancia Fréchet Moderno (FM)	60
5.1.1. Descripción	61
5.1.2. Función costo de FM	62
5.1.3. Etapas previas	63
5.2. Elicitación de Ontologías Naturales	66
5.2.1. Técnica de Redes Semánticas Naturales	68
6. Resultados	79
6.1. Comparación de Ontologías Naturales	79

6.2. Comparación de Imágenes	84
6.2.1. Validación con imágenes	85
6.3. Comparación de Funciones Matemáticas	88
6.3.1. Datos Artificiales I	88
6.3.2. Datos Artificiales II	89
7. Conclusión	93
7.1. Conclusiones	93
7.2. Contribuciones	94
7.3. Publicaciones	95
7.4. Trabajo futuro	95
A. Cuestionario de aplicación	97
B. Formulas Matemáticas	100
C. Herramienta de comparación matricial	102
C.1. Herramienta de comparación matricial	102
C.2. Organización de las carpetas	103
C.3. Comparación de objetos matriciales	104
C.3.1. Ejecución de la herramienta	106
C.4. Detalles Técnicos	107
C.5. Sobre Matlab	108
C.6. Código fuente	109
D. Matrix Maker	123
D.0.1. Funciones evaluadas y rangos de evaluación	123
D.0.2. Funcionamiento de la herramienta	124
D.0.3. Funcionamiento técnico	125
D.1. Código fuente	126

E. Script de validación	142
F. Matrices Artificiales I	145
F.1. Código Fuente	146

Índice de figuras

2.1.	Fruta amarilla y curva: platano	8
2.2.	Modelo de Memoria de Shiffrin y Atkinson	11
2.3.	Ilustración de la estructura hipotética de la memoria de una jerarquía de 3 niveles.	19
2.4.	Diferentes formas de visualización, el criterio más utilizado para escoger alguna, es dependiendo del uso que tendrá.	24
2.5.	Ontología genética: su objetivo es producir un vocabulario dinámico y controlado que pueda ser aplicada a varios campos de la genética .	25
2.6.	Diagrama de las capas que conforma el modelo de Web Semántica .	25
2.7.	Diferentes formas de crear ontologías.	27
2.8.	Redes Semánticas Naturales	30
3.1.	El problema de similitud. Nótese que múltiples criterios de comparación pueden ser usados; volumen, forma geométrica, entre otras; y ninguno de ellos tiene que ser el único.	38
3.2.	El paradigma clásico para el problema de calcular similitud entre entidades. La función de similitud se ha ilustrado como un proceso de toma de decisión binario. Otros casos pueden mapearse al paradigma	40

3.3.	Estructura conceptual del problema de similitud. Entidades a comparar (cuantificables), criterio de comparación (proceso para asignar números en base a alguna regla)	41
3.4.	Criterios para abordar el problema de comparación.	44
3.5.	Relación de equivalencia entre distancia y similitud extraído de [Santos-Camacho, 2014].	44
3.6.	Alineación directa de secuencias entre secuencias C y Q.	47
3.7.	Problema de Similitud Matricial, se consideran las matrices con algún grado de similitud, el problema es cuantificar esa similitud o distancia con un valor escalar.	48
3.8.	Alineación elástica entre las matrices A, B	48
3.9.	Alineación de series de tiempo de forma indirecta, entre la serie C y la serie Q	49
3.10.	Distancia de Fréchet entre dos trayectorias es el tamaño mínimo de la correa que permite el trayecto.	50
3.11.	Matriz de distancias mínimas con un camino trazado, empieza en la posición (1,1) y termina en la posición (p, q).	52
4.1.	Clasificación de técnicas de Alineación Elástica en 2D, para reconocimiento de caracteres escritos a mano (Obtenido de [Uchida y Sakoe, 2005]).	54
5.1.	$R1$ es la fila p acotada en el intervalo $(1, q)$, $C1$ es la columna q acotada en el intervalo $(1, p)$. Similarmente para $R2$ es la fila r acotada en el intervalo $(1, s)$ y $C2$ es la columna s acotada en el intervalo $(1, r)$. Para ambas matrices A y B correspondientes.	62

5.2.	FM es un método de comparación elástica que compara la distorsión que se encuentra en filas y columnas de un elemento de la matriz A sobre todos los elementos de la matriz B, de esta manera se hace un recorrido completo en las dos matrices de entrada.	63
5.3.	Etapas previas de la matriz A: se encuentran en el inciso a, b, c . Mientras que las etapas de la matriz B son los incisos e, f, g . Los incisos d, h son las etapas actuales de cada matriz.	64
5.4.	Esquema del proceso de construcción de RSN.	68
5.5.	Representación matricial de una ontología	78
6.1.	Muestra de las imágenes dentro de la base de datos de imágenes de Princeton.	84
6.2.	Resultados de la técnica Fréchet Moderno utilizando imágenes. . . .	85
6.3.	Organización de la ventana de datos.	88
6.4.	Resultados del experimento de matrices artificiales moviendo ventana de datos	89
6.5.	Algunas matrices del corpus.	90
6.6.	Resultados del experimento de recuperación en una sola función . . .	91
6.7.	Resultado de la búsqueda con elementos mezclados.	92
C.1.	Diagrama General de la herramienta	103
C.2.	Diagrama General de la herramienta	105
C.3.	Diagrama de flujo de la comparación	106
C.4.	Comparación Individual	107
C.5.	Comparación en batch	107
C.6.	Resultados de la comparación en batch usando imágenes	108
C.7.	Resultados de la comparación en batch usando funciones matemáticas	109
D.1.	Interfaz principal de la herramienta Matrix Maker.	125

D.2.	Vista de algunas matrices generadas, notese que se muestra la función, con su respectivo ID y la etiqueta de la funcion con sus parametros.	126
D.3.	Diagrama de contexto del funcionamiento de la herramienta.	127
F.1.	Organización básica de las matrices artificiales	145
F.2.	Matriz de 7x7 llena de ceros con ventana de unos.	147

Índice de tablas

2.1. Significado de la suma en diferentes lenguajes de programación.	8
2.2. Tipos de relaciones semánticas.	23
3.1. Tabla con las restricciones para la distancia DTW.	51
4.1. Tabla comparativa de las propiedades de diferentes técnicas de comparación matricial.	59
5.1. Etapas totales con el cálculo de la función costo correspondiente a cada etapa.	65
5.2. Conceptos usados para el protocolo del dominio de conocimiento de computación	70
5.3. Pesos semánticos para el número de definidores obtenidos.	71
5.4. Lista de definidores de un grupo para un Concepto-estímulo.	73
5.5. Formato de la tabla de captura para el vaciado de los cuestionarios.	73
5.6. Tabla que captura las ocurrencias de los definidores	74
5.7. Tabla que captura las ocurrencias de los definidores	75
5.8. Tabla que captura las ocurrencias de los definidores, convertidas en números.	76
5.9. Tabla que captura las ocurrencias de las definidoras con valores semánticos.	77

5.10. Tabla global de valores semánticos para los conceptos-estímulo	77
6.1. Unión de definidoras. El eje x corresponde a los conceptos definidos, el eje y son las definidoras de los 3 grupos, eje z pesos semánticos . .	81
6.2. Distancias entre grupos.	82
6.3. Unión de definidoras en las matrices SAM	83
6.4. Técnicas de Comparación Matricial	86
6.5. Comparación de los índices de recuperación de diferentes técnicas . .	87
6.6. Matrices de la misma función	91
B.1. Información relevante para la generación de matrices de datos artifi- ciales (I)	100
B.2. Información relevante para la generación de matrices de datos artifi- ciales (II)	101
D.1. Funciones matemáticas evaluadas en los rangos especificados para cada uno de sus parámetros.	124

Capítulo 1

Introducción

En este capítulo se define el problema que se va a abordar, así como aspectos importantes para el desarrollo de la presente trabajo como lo son: objetivos, justificación, contribuciones, descripción breve de la propuesta y por último la estructura de los siguientes capítulos.

1.1. Planteamiento del problema

La comparación de matrices en un nivel muy bajo se refiere al alineamiento elástico de matrices de datos numéricos.

La comparación de ontologías naturales mediante matrices de datos es un caso particular del alineamiento elástico de matrices numéricas. En donde las ontologías son transformadas en matrices. Éste es un problema NP-Completo, el cual además trabaja con entidades que tienen muchos grados de libertad, esto hace que el alineamiento elástico de matrices numéricas sea un problema difícil de tratar.

La comparación de ontologías se presenta como un problema en la cual se va a extraer y comparar conocimiento obtenido de humanos, de allí el nombre de Ontologías Naturales. Se han realizado trabajos para la extracción y almacenamiento

de ontologías o conocimiento humano (elicitación de conocimiento), pero no se puede afirmar que existan métodos que comparen de forma cuantitativa Ontologías Naturales.

Por lo tanto que se ha sugerido emplear una metodología, compuesta una técnica para la elicitación y comparación de conocimiento humano a un bajo nivel, entendiendo bajo nivel como aquella comparación que considere las entidades como un todo, es decir, de manera global.

1.2. Objetivos

1.2.1. Objetivo General

El planteamiento del objetivo general es el siguiente:

- a) Desarrollar y probar una técnica de distancia la cual considere todos los datos en ontologías representadas de manera matricial.

1.2.2. Objetivos Particulares

Se plantean los siguientes objetivos particulares que eventualmente contribuyen y complementan al objetivo general.

1. Desarrollar una metodología de elicitación de ontologías.
2. Comparar conocimiento humano representado en forma matricial.
3. Proponer y probar la comparación de conocimiento obtenido mediante el uso de ontologías naturales.
4. Generar números cuantitativos que midan diferentes aspectos de las ontologías naturales.

5. Validar experimentalmente el desempeño de la técnica propuesta contra algunas técnicas del estado del arte.

1.3. Justificación

En el presente trabajo se presenta una aproximación al problema abierto de alineamiento elástico de matrices. La cual compara directamente la información de matrices numéricas y busca cuantificar el “esfuerzo” de encontrar la distancia mínima entre ellas.

El alineamiento elástico en sí, es un problema NP-Completo, en esta propuesta no es posible identificar una alineación “casilla por casilla”, sin embargo, si es posible cuantificar dicho esfuerzo mediante el alineamiento de las matrices utilizando una **Fréchet Moderno**.

1.4. Contribuciones

- La propuesta de una nueva técnica para realizar la comparación directa entre pares de matrices
- La comparación de conocimiento humano obtenido utilizando Ontologías Naturales
- La comparación canónica de ontologías, es decir, la comparación con diferentes parámetros de una ontología contra un grupo de ontologías.

1.5. Descripción de la propuesta

Definir los requerimientos necesarios para cuando la metodología funciona, así también, mencionar los límites de la técnica.

1. **Obtener las entidades o los fenómenos que se van a comparar:** Las entidades que se pueden comparar pueden ser diversas, siempre y cuando mantengan la propiedad de que se puedan representar con matrices de datos numéricas. Una vez obtenidos los objetos de trabajo, considerar que las matrices no rebasen el tamaño 100 x 100, las matrices no necesariamente tienen el mismo tamaño misma dimensión. En el caso particular de Ontologías Naturales, se deben crear y anotar, este proceso es también llamado **Elicitación**. Es posible utilizar otras entidades, tales como imágenes o matrices de datos artificiales. Cada objeto tiene su proceso de obtención.
2. **Aplicar la comparación de distancia:** Ejecutar la técnica de distancia Fréchet Moderno. El procedimiento de comparación recibe como entrada dos objetos del mismo dominio acotado y obtiene como salida un número escalar que representa la diferencia de los objetos. Este método de comparación responde a las propiedades de métrica que se explicarán más adelante.
3. **Experimentación:** Realizar experimentos con diferentes objetos, esto con el objetivo de corroborar experimentalmente la distancia obtenida. Un experimento posible es crear matrices a partir de funciones matemáticas conocidas y verificar que los resultados obtenidos sean la representación esperada de la diferencia de las funciones conocidas.

1.6. Organización del documento

En los capítulos 2 y 3 se describen los fundamentos teóricos del trabajo, son aspectos importantes que serán utilizados en capítulos posteriores. El capítulo 2 se concentra en Conocimiento y en la manera de obtener conocimiento humano a partir del uso de las Redes Semánticas Naturales. Dentro del capítulo 3 se presentan algunos aspectos teóricos del problema de distancia, cubriendo distancias en R^1 y

R^2 .

Dentro del capítulo 4, se presenta un análisis de los trabajos relacionados y las técnicas de comparación del estado del arte, éstas servirán como referente para los experimentos del presente trabajo.

El capítulo 5 contiene la propuesta desarrollada, incluye la metodología para la obtención de ontologías naturales y allí mismo se explica la técnica de distancia desarrollada.

Los resultados de la experimentación se encuentran en el capítulo 6, se describen los experimentos realizados y las validaciones correspondientes, aquí se utilizan técnicas del estado del arte para validar el desempeño de la técnica propuesta. El capítulo 7 se encarga de las conclusiones y la discusión del trabajo realizado, también se presentan las contribuciones, las publicaciones hechas y el trabajo a futuro.

Al final del presente trabajo se presentan los anexos utilizados para brindar una mejor explicación durante la exposición del tema, también se adjunta el código fuente de los *toolbox* de Matlab creados para esta investigación. La bibliografía es la última sección del documento.

Capítulo 2

Conocimiento

En este capítulo se abordan temas que brindan el sustento teórico de este trabajo; la memoria humana, el lenguaje, la elicitación, las ontologías y las Redes Semánticas Naturales están relacionadas fuertemente con el tema de conocimiento desde la perspectiva computacional. También se aborda problema de distancia, en específico la distancia elástica.

2.1. Sobre el Conocimiento

¿Cómo le decimos “cosas” a una computadora? Es una pregunta que se han realizado muchos científicos dentro de la computación, es un problema que puede ser aproximado dentro de la representación de conocimiento. Lo importante es que una computadora pueda entender las cosas de la misma manera en que las entendemos nosotros como humanos, es decir, que se tenga el mismo significado de las “cosas”.

El conocimiento puede ser adquirido por una computadora mediante dos perspectivas:

1. Clásica: Tesoros y lexicones.
 - Dado por diccionarios.

- En el diccionario el significado es llamado normativo.

2. Moderna: Redes Semánticas Naturales.

- El significado es natural, es decir, como lo usan los individuos.
- Son el significado que dan las personas.

El problema de la perspectiva clásica es que una palabra puede tener varias definiciones, por ejemplo, la simple palabra “Hola” en el diccionario de la [Real Academia Española, 2016] aparece con tres definiciones, una computadora de alguna manera puede tener acceso a un diccionario como éste; sin embargo, en algún momento debe escoger la definición que mejor se adapte al significado que se esté buscando.

Por otro lado, la segunda perspectiva se enfoca en obtener el significado mediante la semántica, dentro ésta se pueden definir un objeto con palabras que expresen características y relaciones, por ejemplo, en la Fig. 2.1 se define un plátano con las características más notables de esta fruta, sin ser una definición de diccionario; sin embargo nuestra capacidad es suficiente para entender que un plátano es una fruta amarilla y curva. Se usaron características de la fruta para definirla. Además, en la semántica, un objeto, proceso o fenómeno puede ser definido de diferente forma pero significa exactamente lo mismo. Para que se pueda entender con claridad, para obtener el significado semántico es necesario tener un cuerpo de conocimiento común para poder entender las características.

Otro fenómeno común, por ejemplo, en la Tabla 2.1 se muestra la operación suma de dos números, el resultado es guardado en la variable x . Se muestra que esa operación puede ser escrita de diferentes maneras pero el significado es el mismo es interesante porque supone que la codificación de la operación es la que tiene el significado, y algunos lenguajes de programación logran entenderla. En la primera columna de la tabla se encuentra la definición de la operación suma, mientras que en la otra columna, se tiene el lenguaje de programación que es capaz de entender esa

forma de escribir la suma.



Figura 2.1: Fruta amarilla y curva: platano

Tabla 2.1: Significado de la suma en diferentes lenguajes de programación.

Suma	Lenguajes
$x += y$	C, C++, C, Java, Perl, Python, Ruby, PHP, etc.
$x := x + y$	Algol, BCPL, Simula, Algol68, SETL, Pascal, Smalltalk, Modula-2, Ada, Standard ML, Objective Caml, Eiffel, Delphi, Oberon, Dylan, VHDL, etc.
ADD x, y	Intel 8086 Assembly Language
LET X = X + Y	early BASIC
$x = x + y$	most BASIC dialects, Fortran
Set x = x + y	Caché ObjectScript
ADD Y TO X GIVING X	COBOL
(incf x y)	Common Lisp

2.1.1. Conocimiento Humano

El estudio del conocimiento humano ha sido durante mucho tiempo un tema de gran interés para filósofos y científicos, se ha tratado desde los tiempos de Platón y Aristóteles intentando responder preguntas como: ¿Qué es el conocimiento humano?, ¿Dónde se almacena?, ¿Cómo es su estructura?, etc. En la actualidad se tienen aproximaciones a estas preguntas que dan un posible acercamiento al conocimiento real del humano, existen modelos que explican la manera de acomodar u organizar el conocimiento humano, cada uno de estos tiene sus implicaciones y limitaciones. Es necesario aclarar, que estos modelos son creados de manera artificial, por lo tanto no brindan la posibilidad de modelar todas las habilidades del cerebro humano durante la cognición.

El conocimiento humano, hablando en una perspectiva general, se ha intentado modelar utilizando perspectivas que intentan copiar el funcionamiento de la memoria humana real, sus características brindan la posibilidad de almacenar, buscar y extraer conocimiento.

¿Qué es Conocimiento?

Para definir el concepto de Conocimiento vamos a usar la definición de [Skinner, 1964], la cual dice que *conocimiento* “es lo que nos **queda** cuando **olvidamos** lo que aprendimos”. Esta definición es importante porque hace énfasis en el uso y manipulación que hacen los humanos en un proceso cognitivo y que da como resultado la abstracción de elementos de información en un complejo proceso construcción-reconstrucción-deconstrucción; en donde, una gran cantidad puede que se pierda, sin embargo, queda representada o utilizada esa información a otro nivel. Otro aspecto interesante de esta definición es que toma en cuenta tanto el aprendizaje como el olvido en sus formas más generales, pero ante

todo describe que el conocimiento es un proceso derivado de la experiencia individual de la persona, en mucho determinada por su interacción social. En suma, es un proceso de representación de información almacenada, ordenada y sistematizada por el individuo en donde esta tanto la experiencia social como la experiencia individual [Vargas-Medina y Figueroa, 2007].

2.1.2. Modelo de memoria humana

Como se mencionó anteriormente, las redes semánticas brindan un modelo más aproximado para la extracción de conocimiento, por tanto, se realiza una revisión de éstas en los siguientes apartados.

El modelo de memoria humana más aceptado es *Modal Model of Memory* propuesto por [Atkinson y Shiffrin, 1968], este modelo consta de tres bloques principales, memoria a corto plazo, memoria a largo plazo y un bloque de procesos de control, como es mostrado en la Fig. 2.2. El proceso inicia con un estímulo registrado por los órganos sensoriales, pasa casi automáticamente a la memoria de corto plazo, se mantiene allí durante unas milésimas de segundo y después desaparece (volátil), posteriormente el estímulo puede ser llevado a la memoria de largo plazo, que sirve como almacenamiento duradero (no volátil). Durante este ciclo ocurren procesos que controlan el flujo de la información, son llamados Procesos de Control, su función es brindar herramientas de recuperación y transporte de estímulos en las memorias. Al final del ciclo se produce una respuesta, que igualmente es controlada por los procesos de control.

Una persona puede registrar toda la información de un estímulo que entre por sus sentidos, la persona utiliza procesos de control para fijar la atención en una porción específica de la información de entrada y utiliza otro proceso de control para repetir esa información y mantenerla allí por unos segundos. Posteriormente esa persona puede Memorizar esa porción de información para usos posteriores usos. El

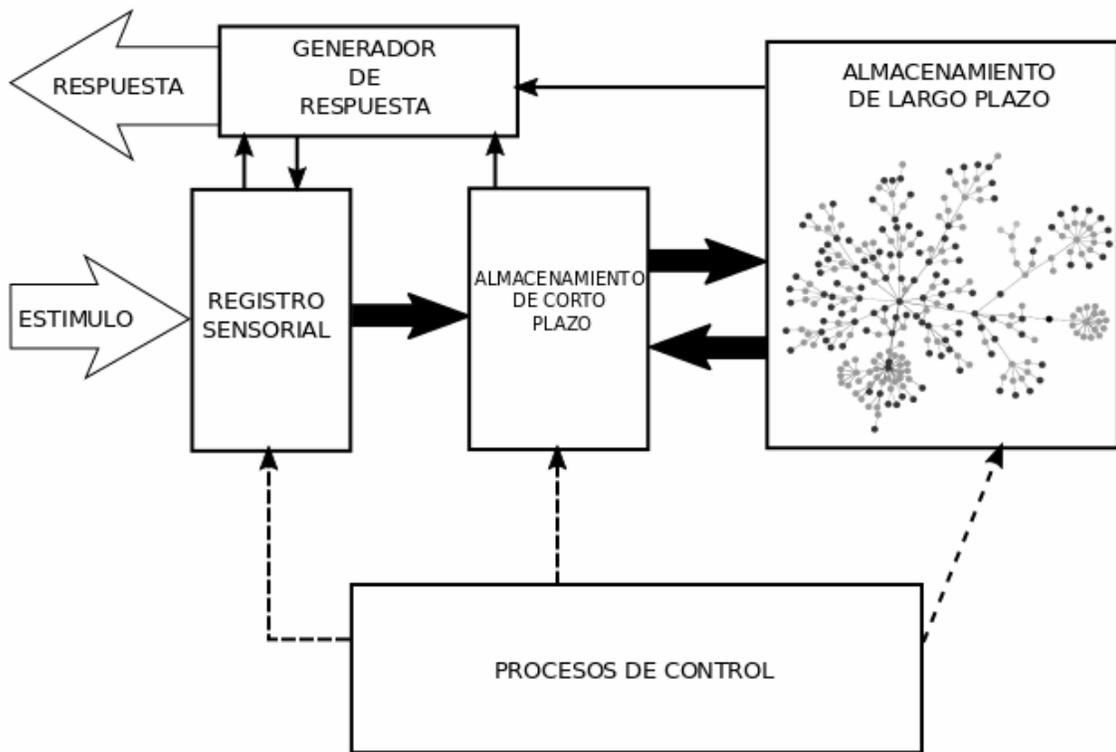


Figura 2.2: Modelo de Memoria de Shiffrin y Atkinson

proceso de memorizar transfiere la información a la memoria a largo plazo donde es almacenada; el proceso de almacenar es llamado: codificación, mientras que el proceso usado para recuperar información almacenada en la memoria de largo plazo es llamado: recuperación. Pertenecen a los Procesos de Control.

Dentro del bloque de memoria a largo plazo, es donde posiblemente se encuentre el conocimiento. La memoria a largo plazo, es también llamada memoria semántica, pues aquí se organizan los conceptos en forma de red, creando relaciones entre los conceptos que un sujeto tenga (vocabulario), a la red de conceptos que se forma se le llama Red Semántica.

2.1.3. Lenguaje

El lenguaje es más que una herramienta de comunicación, funciona con sus ventajas y limitaciones. Para este trabajo el lenguaje se emplea como medio para obtener y entender el conocimiento humano. En este sentido, el lenguaje es la herramienta que utilizamos para modelar la realidad que observamos, Wittgenstein mencionó que el lenguaje explica la realidad y que todo conocimiento de la realidad se encuentra en el lenguaje [Wittgenstein, 1922]. También en su última obra, la idea de que el lenguaje es un acuerdo social es más acertada, es decir, el lenguaje está dado por el consenso de un conjunto de individuos, modelan su realidad en forma de lenguaje [Wittgenstein, 1953]. Esto es muy complejo debido a los diferentes “juegos de lenguaje” que se pueden dar, siempre y cuando se encuentren dentro del mismo conjunto de sujetos para que se puedan entender esos “juegos”.

En Investigaciones Filosóficas [Wittgenstein, 1953], sostiene que el significado de las palabras y el sentido de las proposiciones está en su función, su uso en el lenguaje. Vale decir que preguntar por el significado de una palabra equivale a preguntar cómo se usa. Por otra parte, puesto que dichos usos son muchos, el criterio para determinar el uso correcto de una palabra o de una proposición estará determinado por el contexto al cual pertenezca, que siempre será un reflejo de la forma de vida de los hablantes. Dicho contexto recibe el nombre de juego de lenguaje. Estos juegos de lenguaje no comparten una esencia común sino que mantienen un parecido de familia. De esto se sigue que lo absurdo de una proposición radicarán en usarla fuera del juego de lenguaje que le es propio [Wikipedia, 2016b]. El lenguaje es un instrumento de la vida, por lo que habrá tantos tipos de lenguaje como esferas fundamentales de la vida. Los juegos de lenguaje son los diferentes tipos de lenguaje. El número de tipos de lenguaje no está dado de una vez por todas pues los juegos de lenguaje cambian, nacen nuevos tipos y desaparecen otros [etorredebabel, 2016]. Aterrizando las ideas de Wittgenstein se puede concluir que el significado no tiene esencia y que

el significado no es intrínseco al objeto, si no que existen relaciones que rodean al objeto.

La información que se tiene de la organización de la memoria conjuntamente con lo que sabemos del lenguaje nos ayuda a darle una posible respuesta a las preguntas que se han planteado sobre el conocimiento. Por un lado, el lenguaje nos brinda una manera de modelar la realidad, mientras que la memoria semántica brinda la forma de organización de nuestro conocimiento, de nuestra realidad, con sus respectivas limitaciones.

Relaciones de categorización

El libro [Lakoff, 1987], sirve como *survey* para una gran variedad de estudios empíricos de la naturaleza de la categorización humana. Estos estudios sirven como estándar de la precisión científica en las ciencias cognitivas. Un recuento de estos pasos intermedios inicia con la filosofía de Wittgenstein y avanza a través de la psicología de Eleanor Rosch y su equipo.

No hay nada más básico que la categorización en nuestro pensamiento, percepción, acción y habla. Un entendimiento de cómo categorizamos es central para entender como pensamos y como funcionamos, y más central aún para entender lo que nos hace humanos. Desde los tiempos de Aristóteles hasta los trabajos de Wittgenstein, las categorías fueron bien entendidas y no eran problemáticas: categorías por las propiedades que comparten todos sus elementos. Esta teoría clásica asumía que las categorías fueran como contenedores abstractos, sin cosas dentro y fuera de la categoría. En poco tiempo, todo cambio. La categorización se movió de ser un tema incuestionable a ser la parte central en una gran variedad de disciplinas, debido a sus estudios empíricos. Dentro de la psicología cognitiva, la categorización llegó a ser el principal elemento de estudio, gracias a los trabajos pioneros de Eleanor Rosch, quien hizo de la categorización un problema abierto. Ella se enfocó en dos implicaciones de

la teoría clásica:

- Si las categorías fueran definidas solo por las propiedades que comparten todos sus elementos, entonces ningún miembro debería “ser el mejor ejemplo” de la categoría que cualquier otro miembro.
- Si las categorías fueran definidas solo por propiedades inherentes en sus elementos, entonces las categorías deberían ser independientes a las peculiaridades de cualquier elemento de la Categorización; es decir, no deben implicar a cuestiones tales como la neurofisiología humana, el movimiento del cuerpo humano y las capacidades humanas específicas para percibir, para formar imágenes mentales, de aprender y recordar, para organizar las cosas aprendidas, y para comunicar de manera eficiente.

Rosch observó en estudios realizados por ella misma y su equipo [Rosch y Mervis, 1975], demostraron que las categorías, en general, tienen mejores ejemplos llamados “Prototipos” y que en todas las capacidades específicamente humanas que acabamos de mencionar juegan un papel importante en la Categorización. La idea de que algunos miembros de una categoría pueden ser vistos como “los mejores ejemplos” se llama **Centralidad**. La idea de que al menos algunas categorías tienen niveles de pertenencia y sin fronteras claras se llama **Niveles de Pertenencia**. Uno de los ejemplos excepcionales de experimentación empírica es el trabajo de Paul Ekman, dice que hay emociones humanas básicas y universales que tienen correlación física en expresiones faciales y en el sistema nervioso autónomo. Ekman confirmó ideas tales como conceptos de nivel básico, primacía de nivel básico y centralidad. Al tiempo que demostraba que los conceptos emocionales están materializados. El pensamiento en general está organizado en términos de prototipos y estructuras de bajo nivel, Rosch observó que la categorización en sí misma como uno de los problemas más importantes en cognición y estableció paradigmas de investigación en psicología

cognitiva para demostrar centralidad, parecido entre los miembros, categorización de bajo nivel, primalidad, razonamiento de punto de referencia y ciertos tipos de materialización.

2.2. Elicitación de conocimiento

Tomando las ideas de Wittgenstein se propone una posible manera de organizar el conocimiento en forma de **Red Semántica**, tomando en cuenta las relaciones que se utilizan para definir conceptos. Cabe mencionar que las ideas de Wittgenstein son fuertemente fundamentadas en Filosofía de Lenguaje y abren las puertas para hacer investigaciones de conocimiento en forma de Red Semántica.

La Elicitación de Conocimiento es el proceso mediante el cual se crea un cuerpo de conocimiento acotado a un dominio, este proceso puede ser realizado creando una Ontología. Si este proceso se realiza adecuadamente y de manera cuidadosa, se puede extraer el conocimiento humano. La Elicitación de Conocimiento realizada con **Red Semántica**, en específico la Red Semántica Natural es una aproximación real y mejor establecida por sus bases teóricas en los modelos de memoria. Esta estructura conceptual es rica para trabajar con conocimiento humano real.

¿Qué es Elicitación?

Elicitación (del griego *elicitus*, “inducido” y *elicere*, “atrapar”) es un término de computación que puede referirse más que nada al traspaso de información de un punto a otro, en forma fluida. Está asociado a la psicología como un concepto que refiere al traspaso de información en forma fluida de un ser humano a otro por medio del lenguaje. En computación, dicha asociación es similar, pero la información puede fluir desde un software a otro, de un computador a una persona o de persona a persona. La información posee elicitación si está fluyendo entre los programas, se pueden ver y compartir con otros contactos sin

interrupciones ni dificultades. Cabe aclarar que elicitación es un término que no forma parte del Diccionario de la Real Academia Española y una innecesaria adaptación del término inglés *to elicit* que corresponde a los verbos españoles provocar, suscitar u obtener. Es un modelo de elicitación y representación plausible computacionalmente, que puede ser de gran utilidad en el objetivo de interoperabilidad semántica planteado por la Semantic Web [Wikipedia, 2016a].

2.2.1. Red Semántica

El problema fundamental de la lingüística, la psicología cognitiva y las ciencias de la computación es el problema de significado, en la actualidad una de las corrientes más fuertes que existen para describir el significado son las Redes Semánticas.

Red Semántica

Una Red semántica es una estructura de grafo para representar conocimiento en patrones de nodos y aristas interconectados.

Redes Semánticas (RS) tienen una larga tradición y están especialmente dentro de un método de representación de conocimiento para investigar la semántica. Estas pueden ser rastreadas en los trabajos de Quillian en redes asociativas y son influenciadas fuertemente por sus contribuciones de la psicología cognitiva y el modelado cognitivo.

Implementaciones computacionales de redes semánticas fueron desarrolladas por primera vez por la Inteligencia Artificial y Traducción Computacional, pero versiones teóricas más antiguas han sido utilizadas en la Filosofía, Psicología y Lingüística. De esta manera la Red Semántica está basada en la memoria a largo plazo, brindando una forma de modelar el conocimiento como una serie de relaciones entre conceptos.

El estudio de estas puede ser dividido en dos etapas: antes de Quillian y Collins y después de Quillian y Collins, antes de que ellos publicaran sus trabajos experimentales, se realizaban aspectos teóricos en la investigación filosófica y lingüística, de allí

la distinción realizada. En [Sowa, 2000] se presenta un análisis completo donde se ejemplifica en diversas maneras la idea de “nodos” y relaciones entre nodos en forma jerárquica u organizada, esta misma estructura se puede encontrar desde trabajos de Aristóteles, pasando por Ramón Lulio, Seltz, Fregue, entre otros, en todos esos casos se presentan diferentes esquemas en forma de grafos, donde hay conexiones y nodos, los cuales son el significado de un concepto. La importancia de Quillian y Collins fue demostrar experimentalmente que los humanos posiblemente almacenan información en esta forma (nodos y relaciones de nodos), donde los nodos son conceptos y las relaciones son vínculos con algunas características de los conceptos.

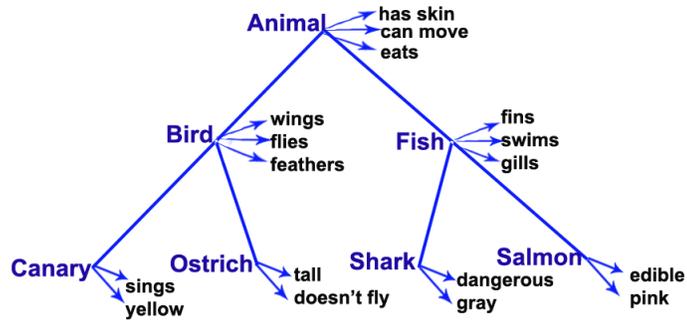
Los orígenes de las Redes Semánticas se encuentran en el trabajo de Richard H. Richens: *Interlingual Machine Translation* [Richens, 1958], propuso un método para traducir lenguajes utilizando diccionarios mecánicos y haciendo comparaciones dentro de éstos para realizar traducción. Richens tomaba en cuenta las ideas acerca de Lenguaje de Wittgenstein en el *Tractatus Logico Philosophicus* [Wittgenstein, 1922], donde se menciona que los lenguajes son secuencias mono-lineales de símbolos. Las Redes Semánticas fueron desarrolladas por Robert F. Simmons [Simmons, 1966] en la Corporativo de Desarrollo de Sistemas (*System Development Corporation*), utilizaba grafos para representar información sintáctica de una palabra y las referencias a los contextos en los cuales es usada, de esa manera Simmons utilizaba redes semánticas para dar el significado a oraciones y tratar de resolver problemas de *Question Answering* [Simmons, 1965]. Más tarde estos trabajos fueron retomados por Collins y sus colegas (Quillian y Loftus). En 1972, Simmons empezó a utilizar modelos computacionales abstractos para representar estas estructuras sintácticas de lenguaje, es decir, utilizaba ya la conjunción entre grafos y diccionarios [Simmons, 1972].

La idea de Redes Semánticas implementa una serie de estructuras de información basadas en una red de conceptos, que en conjunto forman el significado, fue presentada por [Collins y Quillian., 1969] realizando experimentos con los tiempos de reacción, sus

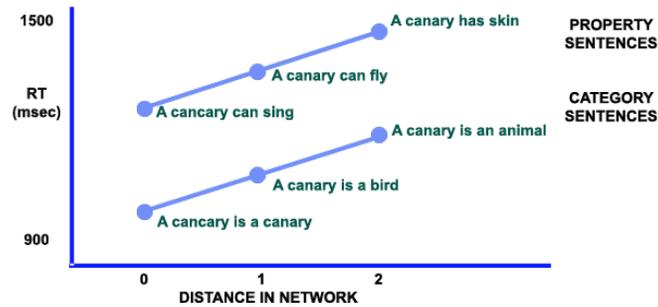
resultados indican que el conocimiento está formado por relaciones entre conceptos. Collins y sus colegas aportaron varios trabajos en el área de las Redes Semánticas para descubrir los principios de diseño para una memoria extensa que permita servir como la base de conocimiento fundamental para el comportamiento lingüístico similar al del humano, se realizaron experimentos con un modelo desarrollado construido dentro de una computadora para “grabar” un cuerpo de información desde un diccionario ordinario a través de una red compleja de elementos y asociaciones interconectadas entre ellas (nodos y relaciones). Los trabajos [Quillian, 1967, Quillian, 1969] propusieron un modelo para almacenar información semántica en la memoria de una computadora. En este modelo cada palabra fue almacenada y consigo una configuración de apuntadores hacia otras palabras en la memoria; esta configuración representa el significado de la palabra. Una organización jerárquica sería almacenar solo la generalización de “Aves” que pueden volar, e inferir de la información almacenada “un canario puede volar”, que un canario es un ave y las aves pueden volar, como es mostrado en la Fig. 2.3a. [Collins y Quillian., 1969] relacionan las redes semánticas con la operación y el funcionamiento de la memoria en humanos. Más tarde, [Collins y Loftus, 1975] presentan una teoría de activación y difusión de procesamiento semántico, el cual fue aplicado a una gran variedad de experimentos bajo esta teoría. La teoría es basada en las investigaciones de memoria semántica de Quillian.

La jerarquía de Redes Semánticas de Quillian y la red de características de Collins y Loftus basada en la proximidad de similitud semántica (ver Fig. 2.3b). Se observan en la Fig. 2.3c donde el código de colores representa cómo la activación se puede esparcir y acumular dado una activación inicial del concepto “fish”. Entre más oscuro representa más activación, esto es acorde a la definición de activación de esparcimiento que se comporta similar.

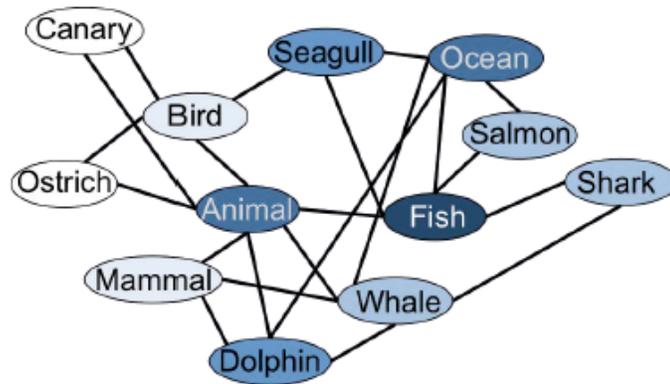
El trabajo de [Meyer y Schvaneveldt, 1976] sirvió en aspectos experimentales



(a) Estructura Jerárquica en conceptos



(b) Distancias entre conceptos



(c) Activación de conceptos en la red

Figura 2.3: Ilustración de la estructura hipotética de la memoria de una jerarquía de 3 niveles.

para que reafirmaran las investigaciones de Quillian, Collins y Loftus: se le presentan dos conceptos a una persona, ésta busca en su red de memoria semántica de forma sistemática para darle significado, empezando en las ubicaciones de los conceptos dados. La búsqueda procede de el primer concepto dado a través de los enlaces o *links* hacia el segundo concepto y de manera iterativa se buscan sobre las categorías sucesivas hasta llegar a una categoría en común. Meyer y Schvaneveldt se dieron cuenta que si los conceptos relacionados son cercanamente almacenados y están conectadas en una Red Semántica, entonces cuando un nodo es activado o “energizado”, esa energía se transmite a los nodos relacionados, proceso llamado **Difusión de Activación**. Esta visión fue muy importante, desde su creación le dieron la apertura en el estudio de la neurociencias por sus aportaciones experimentales, ellos fueron los que abrieron paso a nuevas investigaciones acerca de la organización de la memoria y de su formalización dentro de las ciencias computacionales.

Aunque se pueda modelar interesantes conceptualizaciones de la realidad utilizando Redes Semánticas, es importante conocer una de sus limitaciones, la **capacidad expresiva**, la red, al ser un grafo que conecta nodos y aristas, que pueden conectar millones de nodos es muy probable que su poder expresivo no sea capaz de reflejar lo que está representando, debido a las millones de conexión que se tienen. Esta limitación es de importante relevancia.



El aspecto mas importante de las Redes Semánticas es que funcionan como el modelado basado un conjunto de conceptos y relaciones, estos a su vez forman el Significado. El sustento teórico viene de los trabajos realizados antes de los análisis experimentales de Quillian y Collins, con dichos experimentos se demuestran que son un buen modelo técnico plausible para la organización de conocimiento en humanos.

A partir de los experimentos de Red Semántica se ha intentado implementar y explicar el Significado en diferentes situaciones. Dentro de la Computación, se han realizado esfuerzos enormes para la construcción de **Ontologías**, que son un intento de formular un exhaustivo y riguroso esquema conceptual dentro de un dominio dado, con la finalidad de facilitar la comunicación y el compartir la información entre diferentes sistemas.

2.2.2. Ontología

Una de las representaciones mayormente utilizada en la computación es la Ontología, este modelo une la representación de Red Semántica con la conceptualización de una entidad, ha sido muy poderosa dentro de la Inteligencia Artificial para el intercambio y reuso de información, desde describir los conceptos más generales hasta la descripción de los conceptos contenidos para resolver una tarea [Cristani y Verona, 2005, Dravnieks, 1982, Ashburner *et al.*, 2000].

La Ontología es una rama de Filosofía define el tipo de cosas que existen en el mundo, sin embargo, dentro de la computación no se hace referencia al aspecto filosófico ontológico. Una ontología computacional provee una conceptualización explícita que describe los datos semánticos, proveyendo un entendimiento común y compartido de un dominio. Tom Gruber, fue un científico de la computación en la Universidad de Standford que introduce formalmente el término ontología en ciencias de la computación [Gruber, 1993]:

Ontología: Especificación formal y explícita de una conceptualización compartida.

- **Conceptualización:** modelo abstracto de un fenómeno en el mundo que identifica sus conceptos relevantes.

- **Explícita:** el tipo de conceptos usados y sus limitaciones en su uso, son explícitamente definidas.
- **Formal:** una ontología debe ser computacionalmente legible.
- **Compartida:** una ontología debe capturar el conocimiento consensual aceptado dentro de la comunidad en un dominio específico.

En otras palabras, una ontología es un conjunto de términos de conocimiento, incluyendo el vocabulario, las interconexiones semánticas, y algunas reglas simples de inferencia y lógica para algún dominio en particular [Berners-Lee *et al.*, 2001]. Es decir, un modelo computacional de representación del conocimiento que usualmente es capturado en forma de Red Semántica, una red donde se tienen términos en vocabularios y relaciones entre éstos [Jepsen, 2009].

Una ontología puede representar una variedad de formas, para los sistemas de inteligencia artificial, lo que “existe” es lo que puede ser representado [Genesereth y Nilsson, 1987], pero necesariamente incluirá un vocabulario de términos y alguna especificación de su significado, esto incluye su definición y una indicación de cómo los conceptos son interrelacionados, lo anterior impone una estructura en el dominio y límite de la interpretación de un término [Uschold y Jasper, 1999].

Ontología

Conceptos conectados e interrelacionados en un dominio acotado. Estos conceptos no deben ser estrictamente lingüísticos, si no que pueden ser dibujos, estructuras químicas, figuras, sonidos, olores, etc.

Componentes de una Ontología

Los componentes de una ontología comunmente incluyen [Jakus *et al.*, 2013]:

- Conceptos, clases, colecciones, conjuntos o tipos.

- Objetos, individuos, instancias o entidades.
- Atributos, propiedades o características de los conceptos u objetos.
- Valores de los atributos.
- Relaciones entre objetos.

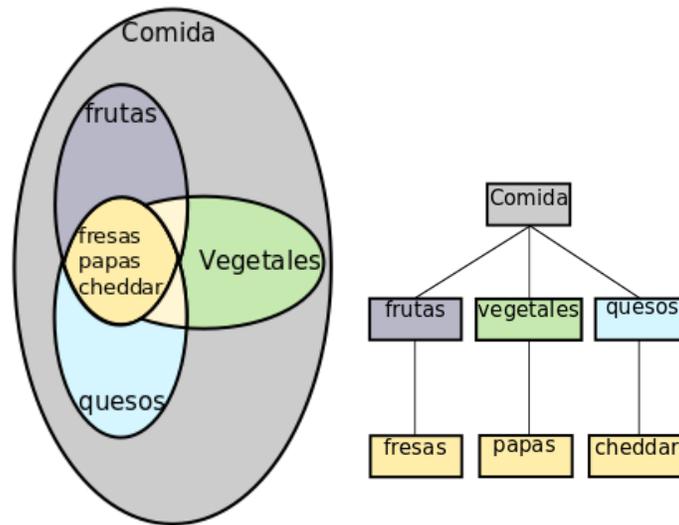
Las ontologías están construidas sobre una taxonomía, que es una estructura jerárquica de conceptos la cual limita la relación entre conceptos a “es un” (*is a*). Sin embargo, con el objetivo de hacer más rica la información es posible agregar relaciones semánticas y componentes adicionales como funciones, restricciones, reglas de inferencia y axiomas. Estos tipos de relaciones adicionales se muestran en la Tabla 2.2:

Tabla 2.2: Tipos de relaciones semánticas.

Relaciones		
Is-a	Instance-of	Has-attribute
A-kind-of	Attribute-of	Object-property
Type-Kind	Has-parts	Object-action
Whole-Part	Connected-to	Action-result
Part-of	Made-of	Object-example

Existen diferentes vistas de una ontología, como se mencionó anteriormente la vista jerárquica que refleja un enfoque de arriba hacia abajo. Otra vista que es ontológica, que representa las propiedades comunes (relaciones semánticas) a través de subclases (ver Fig. 2.8).

Un ejemplo de ontología es la Ontología Genética en la Fig. 2.5: una herramienta para la unificación de la biología [Ashburner *et al.*, 2000]. El objetivo es producir un vocabulario dinámico y controlado que pueda ser aplicada a varios campos de la



(a) Vista Ontológica

(b) Vista jerarquica

Figura 2.4: Diferentes formas de visualización, el criterio más utilizado para escoger alguna, es dependiendo del uso que tendrá.

genética, contribuye a comprender mejor las diversas formas de vida. Con el paso del tiempo se van agregando aún más términos y procesos, este es un ejemplo de Ontología genérica.

Usos de las ontologías

El propósito fundamental para crear y organizar ontologías es el intercambio de conocimiento, sin embargo se utiliza para tareas específicas como intercambio de conocimiento entre agentes de software, en el procesamiento de lenguaje natural, la anotación de medios y la más interesante: **la extracción de conocimiento**.

Para el **intercambio de conocimiento** entre agentes de software se utiliza el modelo de la Web Semántica, donde el standard es utilizar un marco de trabajo de definición de recurso (RDF), este marco es utilizado y forma parte central para la definición del vocabulario de la Web Semántica como se puede ver en la Fig. 2.6.

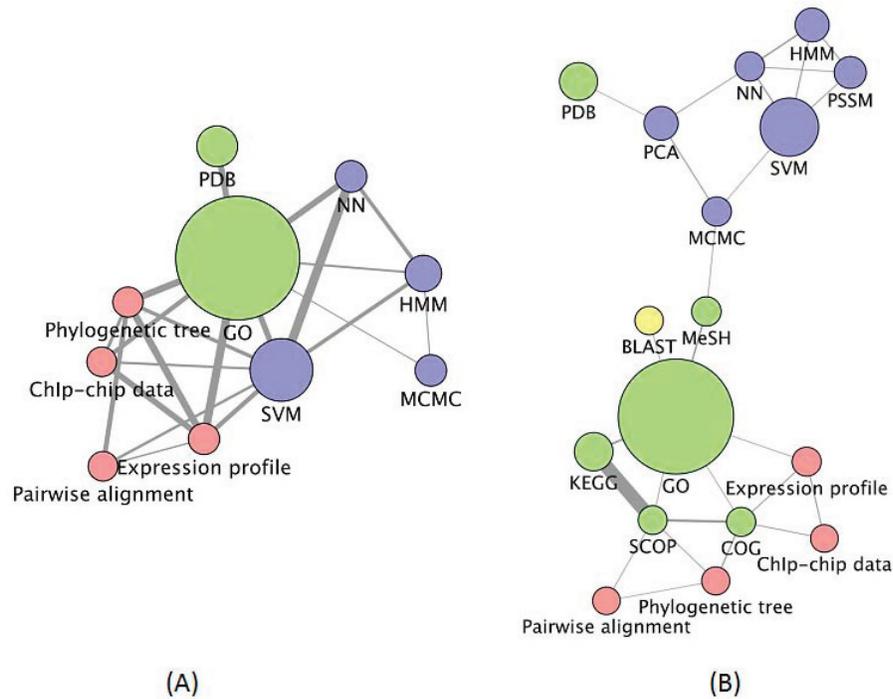


Figura 2.5: Ontología genética: su objetivo es producir un vocabulario dinámico y controlado que pueda ser aplicada a varios campos de la genética

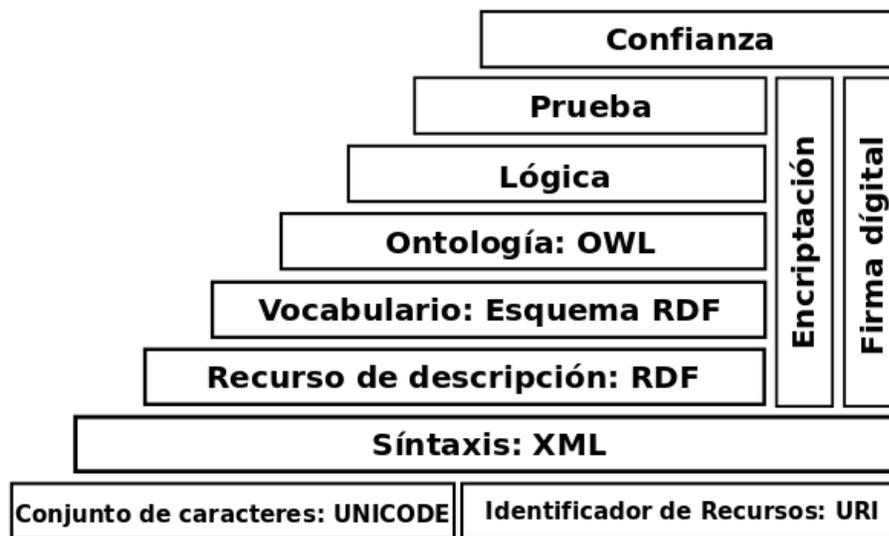


Figura 2.6: Diagrama de las capas que conforma el modelo de Web Semántica

En el alcance del **Procesamiento de Lenguaje Natural**, cuando se habla del significado de las palabras, una ontología debería ser obtenida. Las ontologías son consideradas como un recurso de conocimiento acerca del mundo o de un dominio en específico, consisten de símbolos primitivos usados para la representación de significado. Estos símbolos representan conceptos los cuales son interconectados con una taxonomía, con relaciones de semántica y pragmática de discurso.

Dentro de la anotación de diferentes tipos de **medios** como fotos, audio, vídeos, texto, etc., el propósito primordial es describir el contenido de estos para facilitar el proceso de recuperación en un sistema.

Para la **extracción de conocimiento** se utiliza para obtener conocimiento automáticamente de recursos estructurados o no estructurados. Particularmente en el alcance de este trabajo, nos interesa obtener conocimiento a partir del conocimiento real de humanos. Por lo tanto, las ontologías son una herramienta que puede ser utilizada para este propósito.

Anotación de Ontologías

La Anotación de Ontologías es el proceso de asignación de términos a una ontología y la construcción de relaciones entre conceptos. Esto está relacionado con el tipo de ontología y las herramientas que existen para realizar dicho proceso.

En la actualidad el proceso de construcción de ontologías requiere de diversas herramientas, por ejemplo el uso de lenguajes especiales como OWL (*Ontology Web Language*) y algunas otras específicas como Protege, kaon, Jena, Smore, etc. la mayoría de estas herramientas, en un dominio dado, toman los textos del dominio, ordenan los términos del texto y proporcionan una salida en forma estructurada. El problema central con estas herramientas es que son difíciles de operacionalizar porque utilizan definiciones normativas (ver Fig. 2.7), donde puede existir desacuerdo [Guarino, 1998].

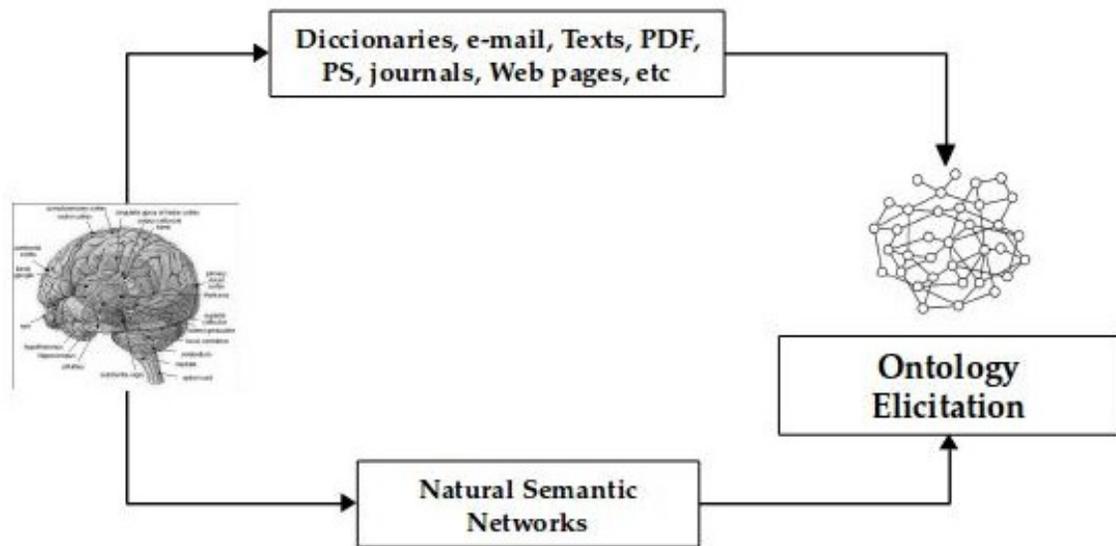


Figura 2.7: Diferentes formas de crear ontologías.

Sin embargo lo más importante es la obtención de la información para la construcción de la ontología. Al proceso específico de construcción de ontologías basado en conocimiento humano se le llama **Anotación de Ontologías** y una gran parte es totalmente manual [C. Bustillo-Hernández, 2009].

En particular para la realización de este trabajo, las ontologías son anotadas utilizando Redes Semánticas Naturales, una herramienta para la obtención de conocimiento real humano.

Ontología Natural

Las Ontologías Naturales corresponden a la definición de Ontología pero son debidamente creadas utilizando las redes semánticas naturales. Son por lo tanto un cuerpo de conocimiento en el cual se tienen conceptos y relaciones entre ellos, definidos para un dominio acotado. Debido a que las Redes Semánticas Naturales se basan en un consenso y en la consistencia de significado.

2.2.3. Redes Semánticas Naturales

Las Redes Semánticas Naturales (RSN) propuestas por [Figuroa *et al.*, 1976], son una manera de obtener conocimiento humano a partir personas. En la actualidad se tienen diversas investigaciones donde se muestra la potencialidad teórica y operativa de las RSN en diversas temáticas [Cabalín-Silva y Navarro-Hernández, 2008, García-Cabrero y Jiménez-Vidal, 1996, Figuroa y Carrasco, 1980, Petra-micu *et al.*, 2012].

Red Semántica Natural

Es una forma de representación de conocimiento, la cual mantiene las relaciones entre los procesos de adquisición y olvido de información. Los procedimientos de las RSN pueden facilitar el proceso de creación de Ontologías y su uso debido a que se tienen valores cuantitativos y no situaciones arbitrarias para establecer la relación entre conceptos.

Las RSN son planteadas como una alternativa de la evaluación de significado, están fundamentadas en los modelos para explicar la forma en la que se organiza la información entorno a la memoria [Quillian, 1967, Quillian, 1969, Collins y Quillian., 1969, Collins y Loftus, 1975, Meyer y Schvaneveldt, 1976]. En esta propuesta también se toma en cuenta que las Redes Semánticas debían ser “Naturales”, donde las redes eran **generadas por sujetos** y no por redes elaboradas en computadora. Esto fue posible gracias a que se retomaron postulados básicos de la organización interna de la memoria a largo plazo, donde la información es contenida en forma de red, los conceptos o eventos forman relaciones, **esas relaciones en conjunto forman el significado de un concepto**. Esto se logra gracias a un proceso reconstructivo de información en la memoria, que permite observar cual es el conocimiento que se tiene de un concepto gracias a sus relaciones [Figuroa *et al.*, 1976]. Es importante considerar la Distancia Semántica, es decir, los elementos que componen la red se encuentran separados en alguna manera, incluso permiten hacer predicciones

[Quillian, 1969], no todos los conceptos son igual de importantes para definir al concepto central. Las RSN están fundamentadas sólidamente por la investigación en memoria humana con visión cognitiva, así también con una visión social del desarrollo de experiencia de los sujetos. En general, la idea de memoria semántica trata de describir la riqueza de las relaciones que se tienen en la memoria humana, estas redes que se van formando con la experiencia del sujeto determinan el significado de los conceptos, donde puede ser posible que un concepto sea definido por algunos, pero ese mismo sea un concepto definidor para otros, es decir, los conceptos de las redes, a su vez también pueden ser conceptos definidores al mismo tiempo, dependiendo de qué parte de la red se active.

En la Fig. 2.8a se muestra la Red Semántica como proceso de interpretación teórica y no simplemente como un grafo de relaciones. Uno de los aspectos más críticos de la investigación en RSN es que se ha confundido con una metodología para graficar conceptos, cuando desde su principio es una teoría de representación de información, que no necesariamente es un grafo [Vargas-Medina y Figueroa, 2007].

Los conceptos para este trabajo son considerados como:

Concepto: Categorías Lingüísticas de Análisis
(que en algunos momentos no necesariamente son lingüísticas, sino que pueden ser imágenes, etc)

Es decir, no son una etiqueta verbal que se defina, sino que las Categorías de Análisis implican más cosas y son más poderosas que una etiqueta verbal. Este aspecto es importante ya que como modelo de representación puede usar todo conocimiento de la psicología cognitiva en forma operacional, como es el ejemplo de la Fig. 2.8b en donde se está representando una red semántica con imágenes, la cual está haciendo referencia directa a la investigación cognitiva contemporánea, que ha demostrado

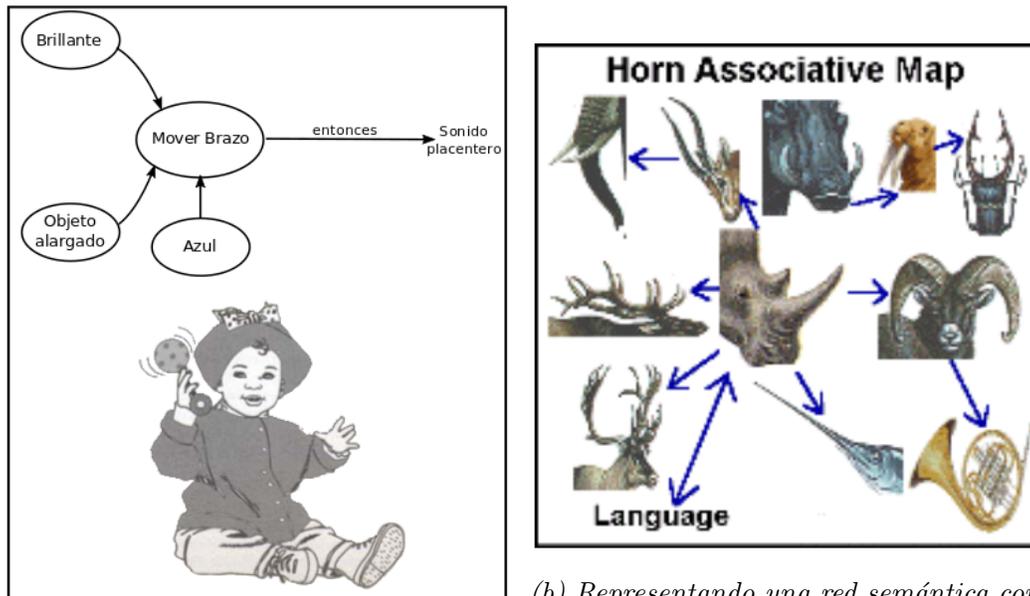


Figura 2.8: Redes Semánticas Naturales

en forma clara, la idea de 2.1.3 prototipicidad de Rosch, donde sabemos por estudios experimentales rigurosos, que los sujetos pueden tener una representación abstracta de imágenes mediante prototipos y no solamente como fotos internas [Vargas-Medina y Figueroa, 2007]. Hay muchas situaciones donde la teoría de RSN tiene un poder descriptivo muy amplio. Por ejemplo experimentos realizados con olores donde son caracterizados por cerca de 150 sujetos [Dravnieks, 1982], los cuales utilizaron una lista de 146 descriptores. Perfiles duplicados altamente correlacionados y consistentemente mayor que los perfiles de diferentes olores. Los perfiles también concuerdan con los obtenidos anteriormente, así los perfiles basados en respuestas compartidas de varios sujetos son construcciones estables y consistentes.

El problema de las Redes Semánticas, tanto las Naturales como las obtenidas por herramientas computacionales (protege) es la **consistencia**. Se repita el proceso de creación y se obtenga el mismo resultado.

Las RSN son **consistentes** cuando las relaciones conceptuales asignadas por un grupo son asignadas con misma jerarquía en repetidas ocasiones [Pérez-Corona *et al.*, 2011, Pérez-Corona *et al.*, 2012]. De esta manera habrán relaciones que se mantendrán para el grupo, quiere decir que las relaciones son las mismas y por lo tanto tienen un peso semántico mayor. Pero si no hay consistencia se habla de un grupo diferente, que no mantienen las mismas relaciones. Esto puede ocurrir como en el ejemplo de los olores, así también con perfumes o melodías.

Consistencia

La metodología de las RSN permite obtener conceptos que se repiten en los diferentes grupos de estudio. Cuando este procedimiento se aplica en forma sistemática a diferentes grupos o en diferentes momentos de tiempo se puede demostrar la alta consistencia de los resultados (mismas jerarquías).

En los experimentos de Quillian y Collins demuestran consistencia de la distancia en tiempo de reacción para alguna cantidad de sujetos. La consistencia se encuentra desde estos experimentos, pero es con el uso de las RSN obtenerla de manera explícita.

Metodología de obtención de las RSN

Para la obtención de las Redes Semánticas Naturales, se consideran dos instrucciones básicas que surgen a partir de las consideraciones anteriores:

- Se pide a sujetos que generen una lista de palabras (conceptos definidores) que definan un concepto (concepto definido).
- Esos conceptos definidores deberán ser jerarquizadas por los sujetos.

En otras palabras, se le proporciona a un sujeto un concepto (nodo central), el sujeto debe proporcionar una lista de conceptos que definan al nodo central, posteriormente a cada uno de esos conceptos definidores el sujeto debe asignarles

una jerarquía con base en la importancia que el sujeto considere a cada concepto definidor.

Con este experimento es posible obtener una red representativa de la organización y la distancia que tiene la información de una persona, obtenida a un nivel de memoria semántica y con ello, el significado de un concepto. El tipo de información que se obtiene gracias a este método, es muy amplio, que ya el sujeto puede proporcionar conceptos de cualquier índole, es decir, la información que se obtiene no se refiere únicamente a los objetos concretos, si no que contiene también eventos, relaciones lógicas de tiempo, afectivas, etc. que permitan tener una idea clara de la representación que tiene la información en la memoria del sujeto, esto justamente hace referencia al significado que tiene un concepto en particular, así lo mencionan en [Figuroa *et al.*, 1976].

La información obtenida con este método, tiene una garantía, se tiene la seguridad de que los datos que se obtuvieron se refieren directamente al significado de un concepto (nodo central) que está siendo definido por los sujetos. Se debe aclarar que las Redes Semánticas, al igual que otros métodos (asociación libre, diferencial semántico), trabajan con materiales verbales, lo cual se ha presentado como una gran limitante para la técnica, ya que es muy posible que las redes semánticas tengan alguna relación con imágenes o algún otro tipo de eventos cognitivos.

Las Redes Semánticas Naturales son una evidencia empírica para el estudio de conocimiento y significado debido a su base teórica, de aquí la importancia que tienen los experimentos realizados por Quillian y Collins, han sido fundamentales para las suposiciones de estructuración del conocimiento en la memoria.

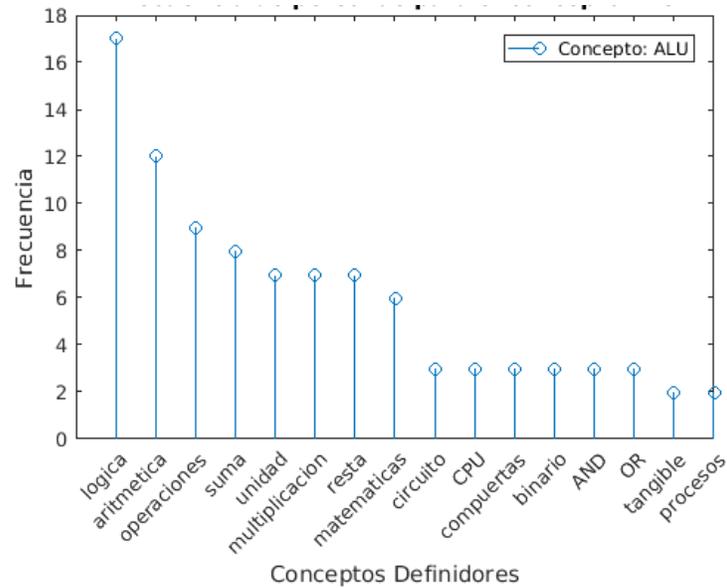


La metodología de obtención de las Redes Semánticas Naturales no es una metodología estadística, sino que usa medidas frecuentistas y no utiliza promedios que de alguna forma suponen datos acomodados de manera gaussiana o alguna otra distribución. Por lo tanto se toman los valores “crudos” de la frecuencia de aparición de los conceptos definidores.

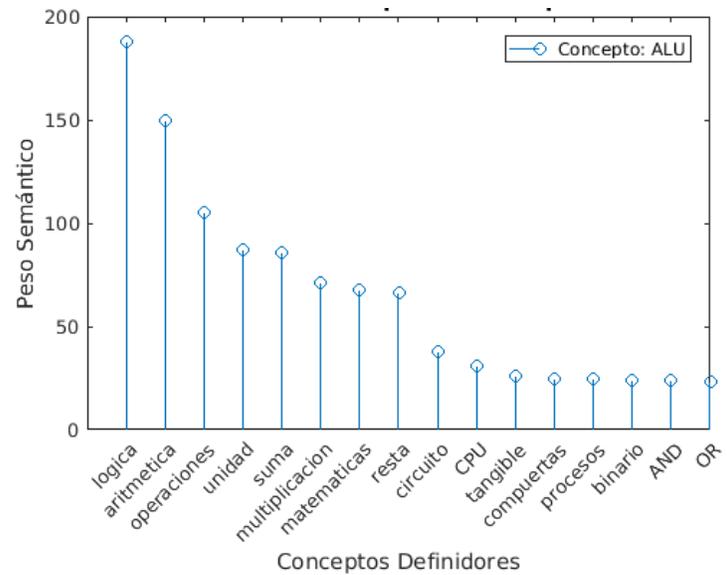
Las medidas obtenidas de las RSN (Valor J, Valor M, VMT, SAM, FMG) son medidas cuantitativas frecuentistas. Estas son una forma de distancia relativa dada por los datos de repetición.

La definición de medida de [Krantz *et al.*, 1971, Krantz *et al.*, 1989, Luce *et al.*, 2006]: *Medir es el acto o proceso de asignar números al fenómeno de acuerdo a alguna regla*, es una buena forma para utilizarse en datos frecuentistas que recaen dentro de un rango. Hay otras escalas de medida, la escala ordinal, la escala nominal, la escala intervalar y la escala de razón [Stevens, 1946], estas escalas tienen sus problemas y las que se consideran buenas medidas son las que tienen Cero Absoluto, pero éstas son muy escasas. Por eso es que se utiliza la definición anterior y trabajar sobre un análisis frecuentista, donde se hace una medida de rango con esto se pueden hacer ciertas operaciones, pero promedios no.

Es importante notar que las medidas M, J, VMT, SAM, FMG brindan una medición simple, por ejemplo en la Fig. 2.9a se tiene la frecuencia de sujetos que utilizaron ciertos conceptos definidores, con operaciones sencillas como una multiplicación se pueden obtener un Peso Semántico que se encuentra en la Fig 2.9b. Así con el peso semántico se puede observar la distancia entre los conceptos definidores. Estas medidas se utilizan en los capítulos 5 y 6, y también se explican a detalle.



(a) Gráfica que muestra la frecuencia de repetición de los definidores del concepto ALU.



(b) Gráfica que muestra el peso semántico de los definidores del concepto ALU dada la repetición anterior.

2.2.4. Otras formas de representación basadas en memoria

Existen además otras formas de representación basadas en memoria humana y matrices, *Latent Semantic Index* es una técnica poderosa al igual que *Latent Dirichlet Allocation* (LDA) que obtienen estructuras de conocimiento dentro de la Inteligencia Artificial. Es importante recalcar que en base a experimentos se justifica que estos trabajos son de memoria humana.

LSA tiene el enfoque de tomar ventaja a las estructuras de alto orden implícitas en la asociación de términos con documentos (estructura semántica) para mejorar la detección de documentos relevantes en la base de términos encontrados en consultas. La técnica particular usada es *Singular Value Decomposition* (SVD), en la cual una matriz de términos contra documentos es descompuesta en un conjunto de 100 factores ortogonales desde los cuales la matriz original puede ser aproximada por combinación lineal. Los documentos son representados por 100 objetos vector de pesos.

Aprendiendo vocabulario desde texto es un caso especialmente conveniente para la investigación. Una nueva teoría general de adquisición de similitud y representación de conocimiento, *Latent Semantic Analysis* (LSA) [Landauer *et al.*, 1987, Landauer y Dumais, 1997, Landauer *et al.*, 1990] es usada para simular satisfactoriamente tal aprendizaje y muchos otros fenómenos psico-lingüísticos. Induciendo conocimiento global indirectamente desde co-ocurrencia de datos locales en un gran cuerpo de texto representativo, LSA adquiere conocimiento acerca del vocabulario completo del idioma Inglés a un vocabulario de menor grado como el que tienen niños de kinder. LSA no usa conocimiento previo lingüístico o conocimiento perceptible similar, es basado solamente en un método de aprendizaje matemático general que obtiene efectos inductivos poderosos extrayendo el número correcto de dimensiones para representar objetos y contextos. Relaciones a otras teorías, fenómenos y problemas son tratados. En LSA el significado de una palabra es definida por su relación con

otras palabras. Así el significado de una palabra es definida como un conjunto de símbolos abstractos (vectores). Esto es similar a los modelos tempranos de [Quillian, 1967] [Collins y Loftus, 1975] en los cuales se define similitud como asociaciones entre objetos individuales, LSA representa esas asociaciones como probabilidades de co-ocurrencias en el texto.

LDA es un método probabilístico para colección de datos discretos tal como un corpus de texto [Blei *et al.*, 2003]. En el cual cada objeto de una colección es modelado como una mezcla finita sobre un determinado conjunto de temáticas. Cada temática es, modelado como una mezcla infinita sobre un conjunto de probabilidades de tema. En el contexto del modelado de texto, las probabilidades de tópicos proveen una representación explícita de un documento. Los significados de una palabra no son almacenados en un lexicón pero son generados en el contexto de la memoria de trabajo (memoria a largo plazo) que registra nuestra experiencia con palabras. Los modelos actuales estadísticos de semántica, tal como LSA, y el modelo de tópicos, describen que es almacenado en la memoria a largo plazo.

Capítulo 3

Distancia

Cuando nos preguntan si podemos identificar si dos objetos son similares, una de las primeras aproximaciones que hacemos es considerar las características particulares de cada objeto y verificar si los dos objetos tienen dichas características, otra aproximación es observar la composición geométrica, sin embargo para estas dos aproximaciones solo se consideran características importantes o sobresalientes para realizar la comparación; esta selección de características significa tomar decisiones que en ocasiones son erróneas y siempre implican una pérdida de información sobre los objetos si se vieran de forma global. Para muchos casos prácticos es mejor utilizar toda la información disponible sobre los objetos involucrados.

Las capacidades del ser humano son sorprendentes, tal es el caso de la capacidad de Abstracción, con ésta se pueden ligar fenómenos que de alguna manera están lejanos, sin algún tipo de relación. El ser humano utiliza múltiples criterios y mecanismos para determinar relaciones, en muchas ocasiones estos criterios son desconocidos, por lo tanto, crear relaciones es un aspecto importante de la inteligencia humana.

Para los humanos el problema de similitud puede parecer muy simple, sin embargo no lo es. Una persona es capaz de reconocer el parecido entre diferentes fenómenos, tal vez puede decir si dos piezas musicales son similares, puede además reconocer eventos

similares del pasado, etc, todas estas actividades pueden ser realizadas identificando las características que a juicio de la persona hacen coincidentes entre los fenómenos. Todo esto se logra gracias a procesos físicos dentro del cerebro no identificados hasta la fecha [Angeles-Yreta, 2006].

El problema de similitud tiene la dificultad de determinar alguna forma de error dentro de los métodos de comparación (Fig. 3.1). Se debe obtener un punto de referencia mediante el cual exista de antemano una clasificación, un orden preestablecido o un contexto acotado [Angeles-Yreta, 2006].

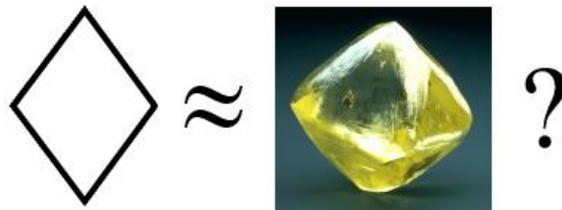


Figura 3.1: El problema de similitud. Nótese que múltiples criterios de comparación pueden ser usados; volumen, forma geométrica, entre otras; y ninguno de ellos tiene que ser el único.

Dentro de la computación, el problema de similitud también es importante porque es muy utilizado en Reconocimiento de Patrones, Minería de datos, Reconocimiento de voz, Procesamiento de Lenguaje Natural, entre otras. El problema de similitud es difícil de abordar dentro de la computación, debido a que la capacidad conceptual de la computadora para establecer relaciones de semejanza entre objetos esta dada por representaciones geométricas, pero no existe ninguna evidencia de que dicho mecanismo sea ocupado por nuestro sistema de referencia: el ser humano. Lo anterior dicho quiere decir, que existen criterios y mecanismos distintos que los seres humanos emplean para determinar relaciones de similitud, pero estos criterios y/o mecanismos tienden a ser muchos y en algunos casos son desconocidos. Estos mecanismos suelen ser diferentes para distintas personas, no existen criterios universales para determinar relaciones de semejanza.

Con lo mencionado hasta ahora se puede decir que la capacidad de generar relaciones entre entidades en los seres humanos es una capacidad fundamental [Ortega-González, 2007].

3.1. Definición de Similitud

Similitud es la cantidad que refleja la fuerza de relación entre dos objetos o dos características [Teknomo, Kardi, 2016].

Existen dos perspectivas sobre la medición de similitud:

- **Paradigma convencional:** La medición de similitud implica la selección de características sobresalientes, obtener éstas supone tomar decisiones erróneas y en ocasiones presentan pérdida de información [Figuroa *et al.*, 2008].
- **Paradigma global:** Emplean la información disponible sobre las entidades a comparar. Para el problema de comparación de matrices, es preferible que los métodos de comparación consideren cada uno de los elementos de los arreglos bidimensionales.

Importancia del problema de similitud

El problema de similitud resulta bastante útil, se utiliza tanto que es muy común que se ocupe en bolsa de valores, en medicina (electrocardiogramas, mastografía), psicometría (huellas digitales, reconocimiento de iris, etc), comportamientos de compra similares, rostros similares, mapas similares, perfiles políticos, textos, códigos, etc. Es decir, la similitud se utiliza en rubros tanto científicos, sociales y económicos. Existen múltiples técnicas para evaluar el problema de similitud entre señales, series de tiempo, secuencias, arreglos, vectores, imágenes, matrices, ontologías, redes

semánticas, textos, etc. En particular interés de este trabajo abordar el problema de similitud en ontologías naturales. En [Angeles-Yreta, 2006] se ha propuesto un paradigma para calcular similitud entre entidades, el cual consta de tres pasos que se pueden observar en la Fig. 3.2:

1. Definir las características de los fenómenos a comparar.
2. Definir la función de similitud basándose en las características definidas.
3. Calcular la similitud entre entidades de acuerdo los dos incisos anteriores.

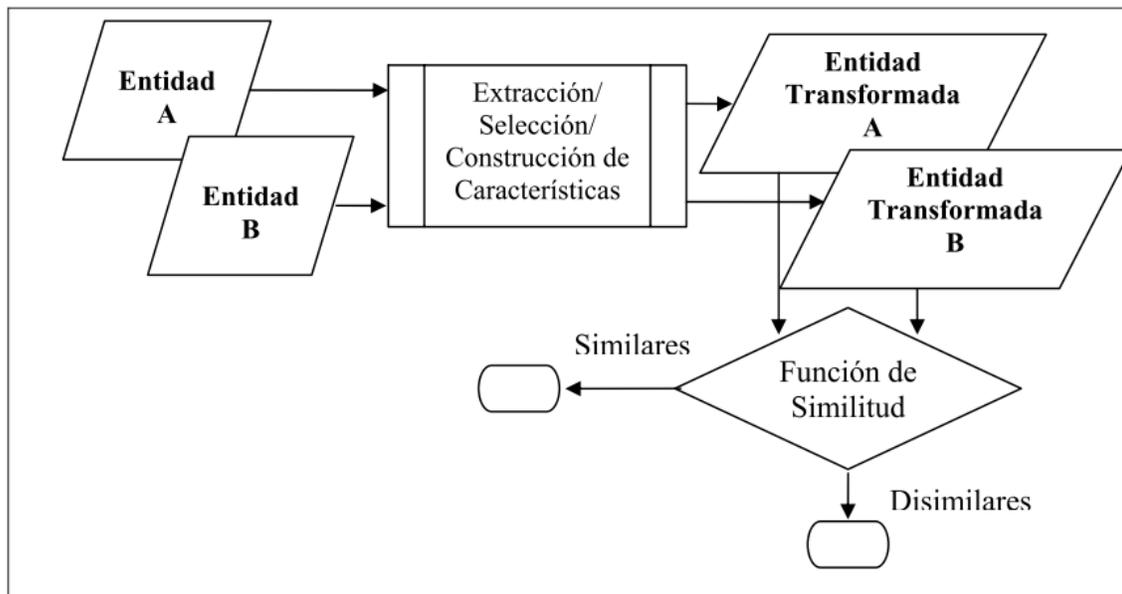


Figura 3.2: El paradigma clásico para el problema de calcular similitud entre entidades. La función de similitud se ha ilustrado como un proceso de toma de decisión binario. Otros casos pueden mapearse al paradigma

Durante la extracción/selección/construcción de características de cada fenómeno se requiere conocer *a priori*, lo significativo del fenómeno en cuestión, así lo que no es considerado significativo es información irrelevante. Este paso es posible que se minimice el costo computacional, debido a que se consideran menos características

en cada fenómeno del proceso de comparación. Los elementos importantes para el problema de similitud son 1) las entidades a comparar, 2) Criterio de comparación, el cual incluye un descriptor de las propiedades que interesan y comparar las entidades como un todo, se observan estos pasos en la Fig. 3.3.

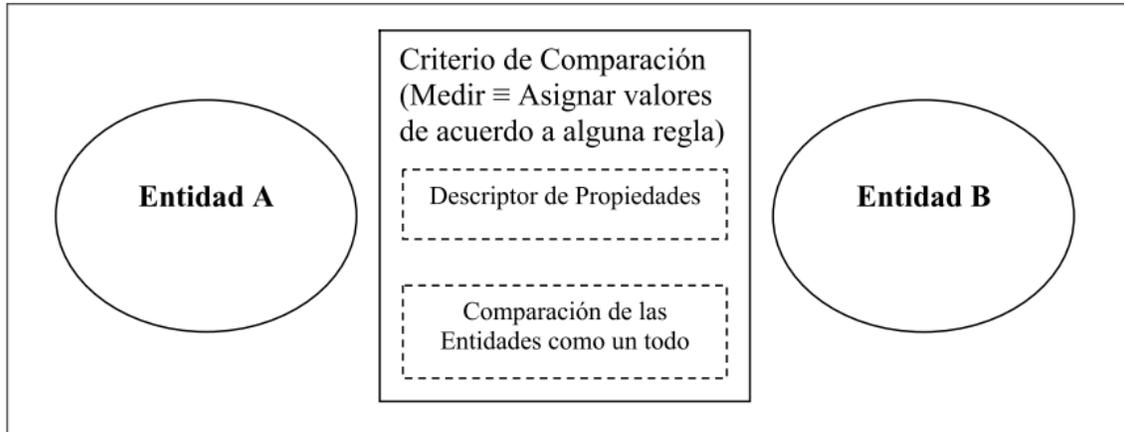


Figura 3.3: Estructura conceptual del problema de similitud. Entidades a comparar (cuantificables), criterio de comparación (proceso para asignar números en base a alguna regla)

- **Entidades a comparar:** El problema de similitud requiere de al menos dos entidades, estas entidades pueden ser arreglos, matrices, grafos, etc. Las entidades deben ser cuantificables, según [Krantz *et al.*, 1971, Krantz *et al.*, 1989, Luce *et al.*, 2006] medir es:

Medir es el acto o proceso de asignar números al fenómeno de acuerdo a alguna regla.

El primer límite para comparar entidades o fenómenos es que el conjunto de estos debe ser **cuantificable**. Todo aquel fenómeno que no pueda ser medido, no puede ser comparado. La teoría de la medida aun sigue siendo estudiada

activamente y se han propuesto funciones de similitud (grado de similitud, razón de similitud, rango de similitud, etc) que ayudan a crear nuevos sistemas de medida.

- **Criterio de comparación:** el criterio de comparación es el acto de medir, la función de similitud del paradigma clásico es un subconjunto de criterios de comparación, el criterio de comparación como un proceso de asignar números de acuerdo a una regla contempla otras formas de medir la similitud entre entidad distintas [Angeles-Yreta, 2006].

Por otro lado, se hace la aclaración de que la validación de las clases o grupos que se pueden generar de alguna técnica de medición, es importante debido a que no existe una clasificación correcta de las cosas *per se*, debido a los diferentes criterios de comparación que se puedan tener sobre las entidades.

- **Descriptor de propiedades:** La función del descriptor de propiedades es representar las entidades a comparar de forma más conveniente según el criterio de comparación, se pueden tener mas de un descriptor para un dominio particular de entidades, no existe un descriptor de propiedades universal, es decir un descriptor que indique las propiedades relevantes de todas las entidades a comparar.
- **Comparación de las entidades como un todo:** un criterio de comparación alternativo es la comparación de entidades completas, estas técnicas con esta característica son llamadas “*gestalt*”.

Las aproximaciones teóricas de Tversky y Lin [Lin, 1998], proponen la descripción de entidades en un conjunto de características definidas, cuando se realizan descripciones muy detalladas hasta el punto de ser imprecisas con su significado real.

Los primeros enfoques para darle sentido al problema de semejanza los propuso [Tversky, 1977], definió que existen cuatro diferentes enfoques para abordar el

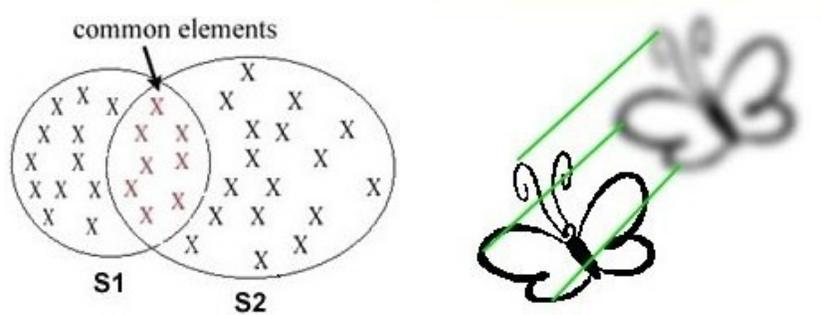
problema:

- a) **Criterio de elementos comunes:** se refiere a la comparación de entidades usando como criterio los elementos comunes (*Common Elements Approach*).
- b) **Criterio de alineación de forma:** es el de realizar comparaciones en donde se persigue emparejar o alinear una entidad con respecto a la otra (*Template Approach*).
- c) **Criterio geométrico:** se refiere a las comparaciones de entidades por su geometría (*Geometric Approach*).
- d) **Criterio por características:** se muestra el enfoque de características es similar al de elementos comunes, aunque no requiere una identificación *a priori* de las propiedades involucradas en la comparación (*Feature Approach*).

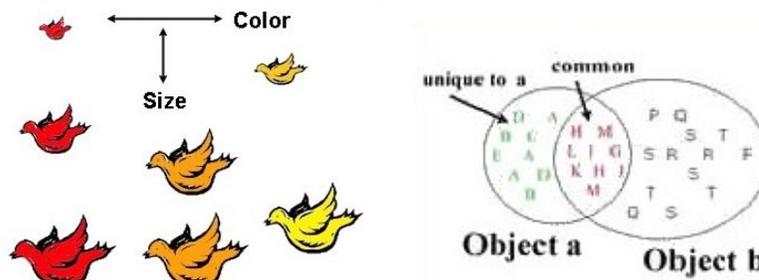
3.1.1. Similitud y Distancia

Como fue mencionado anteriormente, Similitud es una cantidad que refleja la fuerza o intensidad de relación entre dos objetos. La Distancia es la medida de disimilitud entre dos objetos y se refiere a la discrepancia entre dos objetos, basada en varias características analizadas. También puede interpretarse como una medida de desorden en algunos casos, entre dos objetos [Teknomo, Kardi, 2016].

La relación entre distancia y similitud, poca distancia es equivalente a poca diferencia, que es también equivalente a decir que se tiene gran similitud, esto se encuentra ilustrado en la siguiente Fig. 3.5. Con estas equivalencias es válido hablar de una distancia cuando se mide la similitud.



(a) Criterio de elementos comunes (b) Criterio de alineación de forma



(c) Criterio Geométrico (d) Criterio por características

Figura 3.4: Criterios para abordar el problema de comparación.

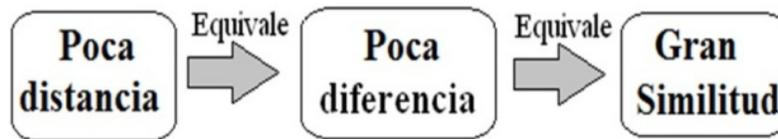


Figura 3.5: Relación de equivalencia entre distancia y similitud extraído de [Santos-Camacho, 2014].

3.2. Espacio Métrico

Un espacio métrico es un conjunto en el cual las distancias entre todos sus miembros son definidos [Zezula *et al.*, 2006]. Formalmente es definido como un par ordenado $M = (D, \delta)$ donde D es un conjunto y δ es una métrica en D , esto es una

función:

$$\delta : D \times D \rightarrow \mathbb{R} \quad (3.2.1)$$

tal que se cumple lo siguiente:

1. No Negatividad: $\forall x, y \in D, d(x, y) \geq 0$
2. Identidad: $\forall x, y \in D, d(x, y) = 0 \Leftrightarrow x = y$
3. Simetría: $\forall x, y \in D, d(x, y) = d(y, x)$
4. Desigualdad Triangular: $\forall x, y, z \in D, d(x, z) \leq d(x, y) + d(y, z)$

La función δ también es llamada función de distancia o simplemente distancia. Comúnmente δ es omitido y uno solo escribe M para un espacio métrico, si éste es claro para el contexto donde la métrica es usada.

En base al postulado de identidad, éste se puede descomponer de la siguiente forma: se obtienen propiedades separadas, Reflexividad: $\forall x \in D, d(x, x) = 0$ y Positividad: $\forall x, y \in D, x \neq y \Rightarrow d(x, y) > 0$.

- Si una función de distancia no cumple la propiedad de positividad, ésta es llamada *pseudo-métrica*.
- Si la propiedad de simetría no se cumple estamos hablando de una *quasi-métrica*.

Las funciones de distancia de espacios métricos representan una manera de cuantificar la cercanía de los objetos de un determinado dominio. A menudo son diseñadas para aplicaciones específicas. Dependiendo del carácter de los valores devueltos, las medidas de distancia pueden ser divididas en dos grupos:

1. **Discretas:** funciones de distancia las cuales solo regresan un conjunto (predefinido) pequeño de valores, por ejemplo la Distancia de Edición.

2. **Continuas:** funciones de distancia en las cuales la cardinalidad del conjunto de valores regresados es muy largo e infinito, la Distancia Euclidiana es un ejemplo.

En la actualidad el problema de distancia es un problema abierto, para R^1 es resuelto con varias técnicas, pero para otras dimensiones R^2 , R^3 , etc, sigue en constante estudio [Deza y Deza, 2009]. Es común que algunas funciones de distancias dentro de la computación no cumplan la propiedad de la desigualdad triangular; sin embargo, son muy utilizadas con buenos resultados para diferentes tareas, tal es el caso de algunas distancias dentro de la teoría de información, como la entropía.

3.3. Distancia en R^1

Comúnmente para utilizar una distancia de una dimensión se utilizan como entidades arreglos, secuencias de elementos o series de tiempo, éstas ultimas pueden representar diversos fenómenos de la naturaleza.

Dadas dos secuencias $C = \{x_i : i = 1, \dots, n\}$ y $Q = \{y_i : i = 1, \dots, n\}$, la distancia entre ellas se remite a cuantificar la dependencia o la independencia entre las secuencias.

En la Fig. 3.6, se tiene la comparación directa de las secuencias C y Q , se puede observar que en las dos secuencias se tiene el mismo patrón, solo que en la secuencia C se tiene un desfase con respecto al patrón en la secuencia Q , el alineamiento realizado no logra identificar el patrón en las secuencias y por lo tanto no minimiza la distancia.

Las medidas de comparación directa son aquellas donde se compara el i -ésimo elemento de la primera secuencia contra el i -ésimo elemento de la segunda secuencia, esta técnica es utilizada en Ciencias de la Computación con diversas aplicaciones y es llamada Distancia Euclidiana, esta distancia es un caso especial de la medida

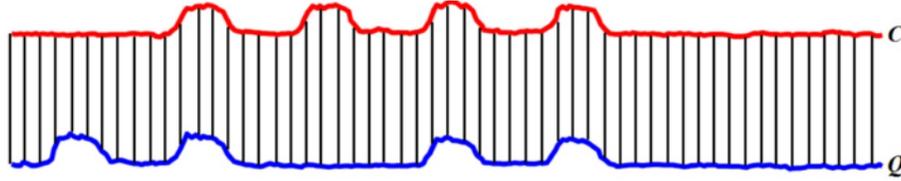


Figura 3.6: Alineación directa de secuencias entre secuencias C y Q .

Minkowski. Pero en muchas otras aplicaciones dentro de la computación, es necesario no hacer una correspondencia directa para encontrar patrones en las secuencias, sino que se busquen los patrones de manera indirecta, es decir, utilizando técnicas de Alineamiento Elástico.

3.4. Distancia en R^2

Asumimos que las secuencias X y Y son matrices y x_i, y_i y x_i, y_i son posiciones de casillas correspondientes en las matrices. Si X y Y representan matrices de 2 dimensiones, las secuencias pueden ser consideradas posiciones en el orden de la exploración de la matriz [Goshtasby, 2012]. La distancia en dos dimensiones sigue siendo importante dentro de la Computación, el tratamiento de imágenes, así como las bases de datos, aprendizaje maquina, etc. siguen realizando importantes aportes dentro de este tema.

El problema de similitud o distancia matricial es importante dentro de la computación y también es difícil de resolver; se debe cuantificar de alguna manera la distancia entre matrices con un valor escalar. Supongamos las 3 matrices de la Fig. 3.7, son la misma matriz pero las últimas dos tienen alguna deformación, podemos decir que son la misma y que difieren en algún grado, el problema radica en encontrar un valor escalar que sea capaz de medir la distancia entre esta matriz.

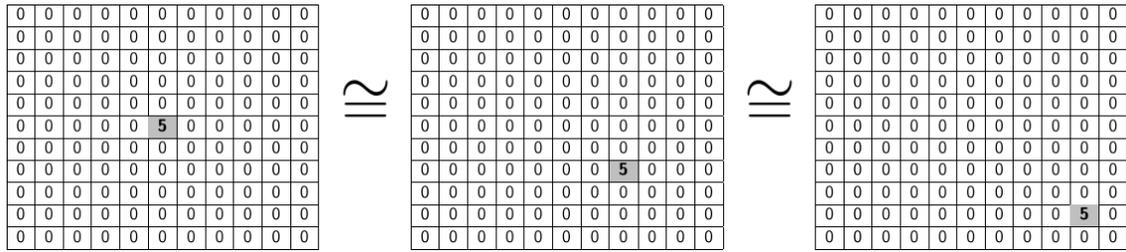


Figura 3.7: Problema de Similitud Matricial, se consideran las matrices con algún grado de similitud, el problema es cuantificar esa similitud o distancia con un valor escalar.

3.5. Alineación Elástica

La Alineación Elástica (*Elastic Matching*) es definido como la optimización al problema de las deformaciones bidimensionales (*Two-dimensional warping*(2DW)), 2DW es un mapeo pixel-pixel entre dos imágenes [Uchida y Sakoe, 2000].

Dadas dos matrices A, B , la alineación elástica es la distancia entre A y B después de “ajustar” B a A lo más cerca posible [Uchida y Sakoe, 2005]. La Fig. 3.8 muestra un ejemplo de el ajuste que debe ser realizado.

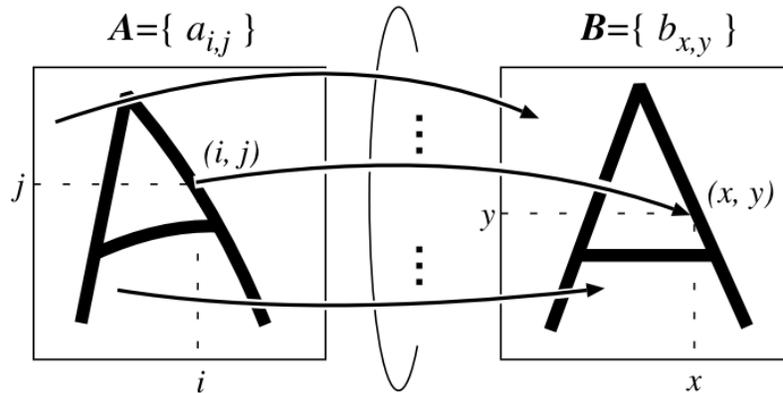


Figura 3.8: Alineación elástica entre las matrices A, B

Técnicas basadas en la alineación elástica miden la cantidad de esfuerzo que se necesita para realizar dicho “ajuste” en las matrices. [Santos-Camacho, 2014] explica que la alineación elástica o comparación elástica es un “valor” que refleja el “esfuerzo”

entre una relación de información contenida en un par de secuencias, como se aprecia en la Fig. 3.9, donde el esfuerzo es un valor cuantitativo de semejanza entre las secuencias comparadas. Es decir, se mide de manera cuantitativa el esfuerzo que se tiene al alinear una secuencia con la otra, el resultado de este proceso es un valor numérico. Este es un problema difícil, en [Keysers y Unger, 2003] se demuestra que la alineación elástica tiene una complejidad NP-Completo.

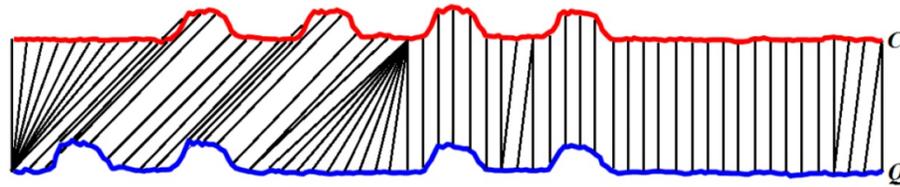


Figura 3.9: Alineación de series de tiempo de forma indirecta, entre la serie C y la serie Q

Es importante mencionar que el problema de alineación elástica nace en trabajos relacionados con el reconocimiento de caracteres escritos a mano, donde imágenes son tomadas como matrices; sin embargo, se puede ocupar para matrices numéricas de otra índole.

3.5.1. Distancia Elástica R^1

La primera aproximación de distancia elástica fue introducida por Maurice Fréchet [Fréchet, 1906], la intuición planteada por Fréchet es simple: supongamos que una persona sale a caminar con su perro, este va atado con una correa, los dos avanzan sobre su propia trayectoria controlando su velocidad y sin poder regresar, la distancia Fréchet es la longitud mas corta de la correa en la trayectoria (Ver Fig. 3.10).

Distancia Dynamic Time Warping

El algoritmo Dynamic Time Warping (DTW) tiene sus orígenes en los fundamentos del reconocimiento de voz [Rabiner y Juang, 1993, Chu *et al.*, 2001], es una función

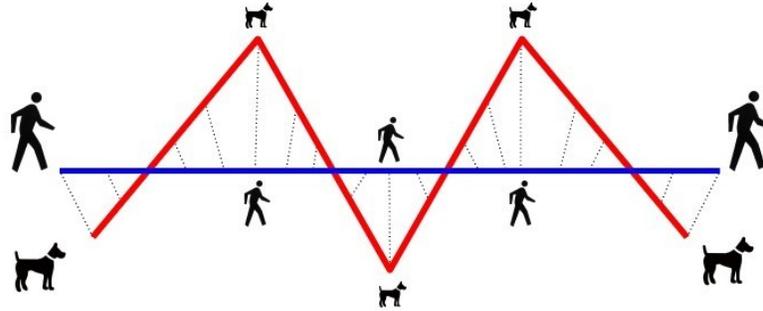


Figura 3.10: Distancia de Fréchet entre dos trayectorias es el tamaño mínimo de la correa que permite el trayecto.

recursiva que minimiza la distancia base entre dos secuencias, comúnmente la distancia base es la distancia Euclidiana, pero puede ser definida flexiblemente según se requiera para la aplicación.

Def: Dadas dos secuencias c, q de longitud p, t respectivamente, denotadas por: $c = \langle c_1, c_2, \dots, c_p \rangle$ y $q = \langle q_1, q_2, \dots, q_t \rangle$ el cálculo de la distancia DTW está dada por resolver la siguiente función de recurrencia:

$$D(i, j) = D_{base}(c_i, q_j) + \min \begin{cases} D(i-1, j) \\ D(i-1, j-1) \\ D(i, j-1) \end{cases} \quad (3.5.1)$$

La distancia base puede ser la Distancia Euclidiana, pero puede ser libremente modificada: $D_{base}(c_i, q_i) = \sqrt{(c_i - q_i)^2}$.

Se puede concluir que si la distancia DTW entre las secuencias c, q tiende a inf , entonces c, q son diferentes, además si DTW de c, q tiende a 0 son iguales. DTW construye una matriz de tamaño $(C \times Q)$ de valores mínimos como en la Fig. 3.11, la tarea es encontrar un camino dentro de la matriz de tal manera que empiece en $(1, 1)$ y termine en (C, Q) así, el costo acumulativo mínimo a través del camino es el requerido. A este camino se le llama *warp path* y se denota con r , es un conjunto de elementos de la matriz generada que definen un mapeo entre c, q ; donde r satisface

las siguientes restricciones:

Tabla 3.1: Tabla con las restricciones para la distancia DTW.

Restricción	Descripción
Inicio/fin	Requiere que el elemento c_1 sea relacionado con q_1 , y el elemento c_p relacionado con q_t , esto garantiza que el camino de la ventana comience y finalice en diagonal.
Monótona	Los elementos en el <i>warp-path</i> deben cumplir con la siguiente condición: r_{k-1} menor que r_k en sus índices (i, j) , esto presiona a los puntos en r a espaciarse de forma uniforme en el tiempo.
Restricciones Locales	Las restricciones locales se utilizan para reducir el rango de búsqueda de los elementos en r , con base a la vecindad de un elemento en la matriz.

Si el camino pasa por la celda (i, j) contribuye al costo acumulativo. Debemos tener en cuenta que existe más de un camino con esas características, pero el que necesitamos es el que nos brinde el mínimo costo [Salvador y Chan, 2007].

DTW usa la distorsión del tiempo por aceleración y/o transformaciones de declaración para obtener un mapeo que minimiza la distancia entre dos secuencias. DTW es una variante de la Distancia Fréchet [Chouakria y Nagabhushan, 2007].

3.5.2. Distancia Elástica R^2

Este trabajo se enfoca a la distancia en R^2 , se explicará con detalle la propuesta de una técnica que sirve como aproximación al problema de distancia en R^2 . Problema de optimización con respecto a un mapeo casilla/casilla, llamado Two dimensional

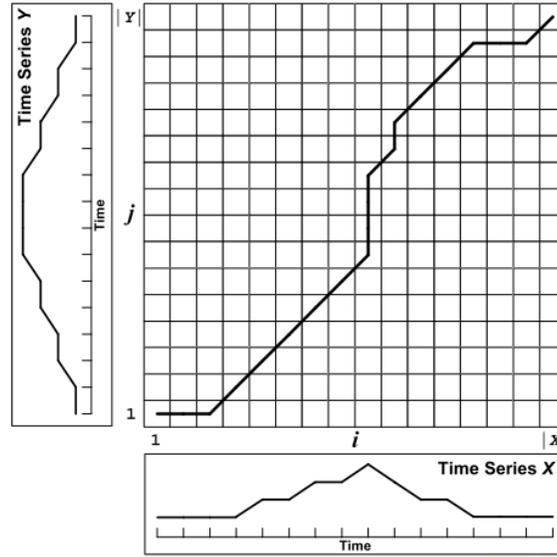


Figura 3.11: Matriz de distancias mínimas con un camino trazado, empieza en la posición $(1,1)$ y termina en la posición (p, q) .

Warping (2DW) en español es Deformación Bidimensional [Lei y Govindaraju, 2004], es generalizado utilizando diferentes datos acomodados en forma de matriz.

Lo anterior requiere realizar una comparación elástica matricial, similar a la alineación elástica para el caso de 1 dimensión (arreglos o secuencias). Las matrices son estructuras muy importantes dentro de las matemáticas y las ciencias de la computación, por el simple hecho de que pueden representar diferentes tipos de información, grafos, redes, ontologías, por decir algunos ejemplos. La similitud entre matrices puede ser excesivamente complicada debido a que el grado de libertad para un arreglo bidimensional llega a ser enorme, por ejemplo, en [Ortega-González, 2007] con matrices en forma de imágenes explica que para una imagen de tamaño 100×100 pixeles, el grado de libertad es de $(256^{100})^2$, donde cada pixel tiene un intervalo entero de valores de $[0 - 255]$.

Capítulo 4

Estado del Arte

La búsqueda en el estado del arte para una técnica de comparación global de matrices basada en Alineamiento Elástico, aplicada en la obtención de distancia entre ontologías naturales.

Dentro del estado del arte no existen técnicas de medición de distancia que realicen dicha comparación, por lo tanto se analizan aquellas técnicas que realicen una comparación cuantitativa de matrices de datos, también se realiza una revisión de técnicas que realizan comparación elástica en R^2 .

4.1. Distancias basadas en Elastic Matching

La búsqueda de técnicas que realicen una comparación elástica se encuentran en el marco de resolver la tarea de reconocimiento de caracteres escritos a mano [Uchida y Sakoe, 2005], los esfuerzos para resolver dicha tarea se limitan a obtener características predefinidas de las imágenes de los caracteres, sin embargo, no realizan una comparación global.

Dentro de la clasificación de técnicas que realizan un comparación elástica en 2D para la tarea de reconocer caracteres escritos a mano, se encuentran las paramétricas y

las no paramétricas. Las paramétricas son las que involucran la entrada de parámetros controlados para la ejecución de la comparación, mientras que las no paramétricas no utilizan dichos parámetros.

Nuestro interés es revisar las técnicas no paramétricas, discretas y restringidas, como se muestra en la clasificación de la Figura 4.1, las técnicas No Paramétricas, discretas y restringidas se pueden optimizar utilizando la técnica de Programación Dinámica; sin embargo [Uchida y Sakoe, 2005] realiza un análisis de diversas formas de aproximación al problema 2DW.

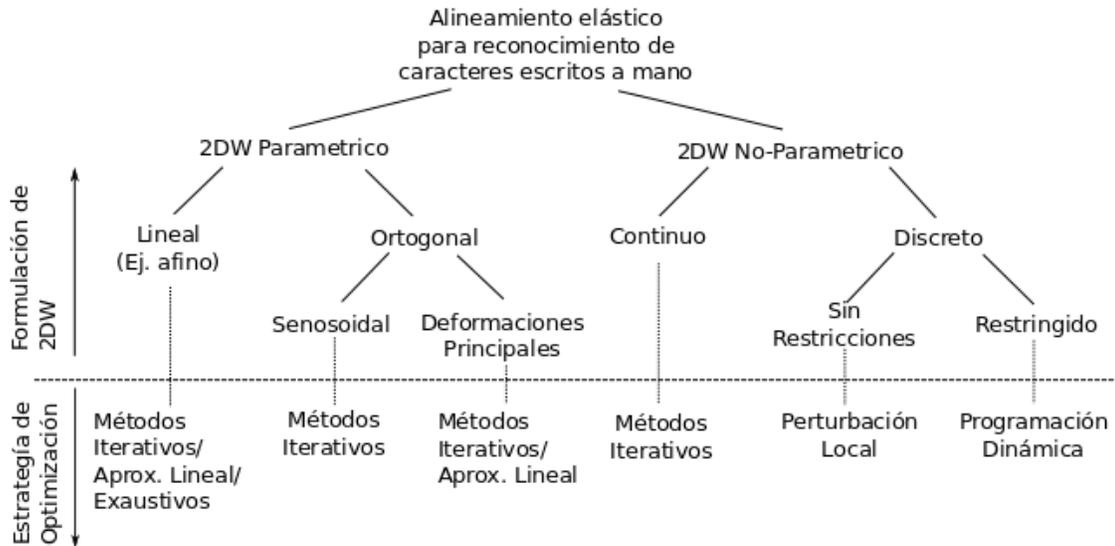


Figura 4.1: Clasificación de técnicas de Alineación Elástica en 2D, para reconocimiento de caracteres escritos a mano (Obtenido de [Uchida y Sakoe, 2005]).

Las técnicas No paramétricas y restringidas tienen una subclasificación (DP1-DP7), la técnica más aceptada es propuesta en [Uchida y Sakoe, 1998], la cual realiza un mapeo de dos dimensiones, monótono y continuo, que es basado en Programación Dinámica.

Considera dos imágenes $A = \{a(i, j)\}(i, j = 1, \dots, N)$ y $B = \{b(x, y)\}(x, y = 1, \dots, M)$. El mapeo óptimo dos dimensional monótono y continuo entre A y B es

definido por la función de alineación $(x(i, j), y(i, j))$ la cual minimiza la siguiente función criterio:

$$\sum_{i=1}^N \sum_{j=1}^N |a(i, j) - b(x(i, j), y(i, j))| \quad (4.1.1)$$

sujeta a las condiciones de monotonicidad y continuidad:

$$0 \leq y(i, j) - y(i, j - 1) \leq 2 \quad (4.1.2)$$

$$|x(i, j) - x(i, j - 1)| \leq 1 \quad (4.1.3)$$

$$0 \leq x(i, j) - x(i - 1, j) \leq 2 \quad (4.1.4)$$

$$|y(i, j) - y(i - 1, j)| \leq 1 \quad (4.1.5)$$

Las relaciones de monotonicidad y continuidad horizontales y verticales entre un pixel y sus cuatro adyacentes son preservados por estas condiciones. Así la estructura topologica es las imagenes es conservada. A estas condiciones se les agrega la condiciones de limite:

$$x(1, j) = y(i, 1), x(N, j) = y(i, N) = M \quad (4.1.6)$$

Sea $D(A, B)$ que denota el criterio minimo de (1), por lo tanto:

$$D(A, B) = \min_{x(i, j), y(i, j)} \sum_{i=1}^N \sum_{j=1}^N |a(i, j) - b(x(i, j), y(i, j))| \quad (4.1.7)$$

4.2. Medidas R^2

Para todas las técnicas de comparación en R^2 se busca que cumpla las propiedades de distancia: sean A y B matrices de datos numéricos, d una función de distancia tal que $d(A, B) = k$, donde k esta en \mathbb{R} .

4.2.1. Distancia Euclidiana

Esta técnica no realiza una comparación elástica, es una comparación directa, por lo tanto no se considera como eficiente. La Euclidiana toma dos matrices de entrada y por salida se obtiene un número k , cumple las características buscadas para nuestro análisis. Es una de las técnicas más básicas para la comparación de estructuras computacionales, la hay para objetos en general de n dimensiones [Chen y Chu, 2005].

La distancia Euclidiana entre x y $y \in R^n$ es calculada por:

$$\delta(x, y) = \|x - y\|_2 = \sqrt{\sum_{j=1}^d (x_j - y_j)^2} \quad (4.2.1)$$

Una medida similar es llamada Distancia Manhattan, la cual ejecuta menos operaciones, es calculada con la siguiente forma:

$$\tau_1(x, y) = \|x - y\|_1 = \sum_{j=1}^d |x_j - y_j| \quad (4.2.2)$$

Otra distancia es llamada la Norma Suprema, la cual se calcula:

$$\tau_2(x, y) = \max_{1 \leq j \leq n} |x_j - y_j| \quad (4.2.3)$$

4.2.2. Distancia Chord

La distancia Chord entre dos vectores x y y es la medida de distancia entre los vectores proyectados de x y y a una esfera unitaria, la cual puede ser calculado con:

$$\delta(x, y) = \left\| \frac{x}{r} - \frac{y}{s} \right\|_2 \quad (4.2.4)$$

donde $r = \|x\|_2$, $s = \|y\|_2$. Un simple cálculo permite que $\delta(x, y) = 2 \sin(\alpha/2)$

4.2.3. Coeficiente de Rango Spearman

Una medida derivada del coeficiente de rango de spearman

$$\gamma(x, y) = 1 - \frac{6 \sum_{j=1}^d r_j^2}{d(d^2 - 1)} \quad (4.2.5)$$

4.2.4. Distancias Estadísticas

Se analizan ocho índices diferentes que son utilizados para el análisis de distancia en imágenes [Vaiopoulos, 2011], cada índice corresponde a una técnica de distancia matricial, su propósito inicial fue la valoración de la calidad en la fidelidad espectral en imágenes. Sin embargo, se estima que pueden ser utilizados para la comparación de matrices de origen completamente diferente. El valor promedio es calculado para cada índice. Los índices se pueden calcular de la siguiente manera: para cada formula, sean A y B dos matrices que tienen las mismas dimensiones, y sean \bar{A} y \bar{B} la media de los datos en A y B respectivamente:

- **Bias:** Es basado en la división de los valores medios de la matriz original (A) y la matriz procesada (B). La idea teórica, el valor ideal de *Bias* es 0, un valor pequeño positivo o negativo en el *Bias* significa que existe una similitud fuerte entre A y B .

$$Bias = 1 - \frac{\bar{B}}{\bar{A}} \quad (4.2.6)$$

- **Coeficiente de Correlación (CC):** define la correlación entre dos matrices. Es considerado un índice preciso y comúnmente utilizado. Tiene un rango dinámico $[-1, 1]$. El valor 0 significa que no hay correlación alguna entre los datos de las matrices, mientras que el valor 1, indica que $A = B$, el valor -1, significa que A es el opuesto exacto de B o $A = -B$.

$$CC = \frac{\sigma_{A,B}}{\sigma_A \sigma_B} \quad (4.2.7)$$

- **DIV (Diferencia en varianza):** representa la varianza de la matriz B dividida entre la varianza de la matriz A, esto restado a 1, la interpretación es igual que en *Bias*:

$$DIV = 1 - \frac{\sigma_B^2}{\sigma_A^2} \quad (4.2.8)$$

- **ERGAS (Erreur Relative Globale Adimensionnelle de Synthèse):** Es el error relativo global sin dimensiones de síntesis, denota el radio del tamaño de pixel entre la pancromática resolución espacial.

$$ERGAS = 100 \frac{h}{l} \sqrt{\sum_{k=1}^N \frac{1}{N} \frac{RMSE(B_k)^2}{\bar{x}_k^2}} \quad (4.2.9)$$

- **Q (Universal image quality index):** Éste índice se refiere a la calidad universal de una imagen, describe la calidad, es un índice robusto, utiliza la media y la desviación estandard de las matrices para producir el resultado. Este índice es calculado globalmente para las matrices. Tiene un rango dinámico como el coeficiente de correlación de $[-1, 1]$.

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2\sigma_y^2)[\bar{x}^2 + \bar{y}^2]} \quad (4.2.10)$$

- **RASE (Relative Average Spectral Error):** este índice utiliza la media global (todas las bandas) de radianza en la matriz original. Debido a que se basa en RMSE, su valor ideal teórico es 0.

$$RASE = \frac{100}{\bar{x}} \sqrt{\frac{1}{N} \sum_{k=1}^N RMSE(B_k)^2} \quad (4.2.11)$$

- **RMSE (Root Mean Squared Error):** es uno de los índices comúnmente más utilizados. Este es el error de la diferencia al cuadrado, de dos conjuntos de datos (x,y), dividido por el número de elementos (pixeles), n:

$$RMSE = \sqrt{\frac{\sum_{k=1}^n (x_i - y_i)^2}{n}} \quad (4.2.12)$$

4.2.5. Comparativa de las técnicas

En el Cuadro 4.1 se presenta un análisis donde se observan criterios de las diferentes técnicas para la comparación matricial. El primer criterio es el tamaño, se refiere a que la técnica considera matrices de diferente tamaño o matrices cuadradas. La columna Escalar indica que la salida producida por la técnica es un escalar, y la siguiente es el rango de ese escalar. El cuarto indicador se refiere a que la técnica necesita de algún parámetro extra para poder operar. En el último criterio se especifica el tipo de la técnica.

Tabla 4.1: Tabla comparativa de las propiedades de diferentes técnicas de comparación matricial.

Técnica	Diferente tamaño	Escalar	Rango	Paramétrica	Tipo
Bias	si	si	$[-1, 1]$	no	Estadística
Correlación	no	si	$[0 - \infty]$	no	Estadística
div	si	si	$[0 - \infty]$	no	Estadística
ergas	no	si	$[0 - \infty]$	si	Estadística
q	no	si	$[0 - \infty]$	no	Estadística
RASE	no	si	$[0 - \infty]$	no	Estadística
RMSE	no	si	$[0 - \infty]$	no	Estadística
Información Mutua	no	si	$[0 - \infty]$	no	Teoría de la Información
Hausdorff	no	si	$[0 - \infty]$	no	Alineación Elástica
Monotonic, Continuos	si	si	$[0 - \infty]$	no	Alineación Elástica
Fréchet Moderno	si	si	$[0 - \infty]$	no	Alineación Elástica

Las técnicas que son estadísticas al igual que las técnicas de teoría de la información tienen la limitación de que, en algún momento realizan operaciones que suelen perder información, como normalizar o promediar los valores. Otras, que también son de la teoría de la información obtienen características.

Capítulo 5

Fréchet Moderno

Este capítulo contiene la metodología propuesta para la comparación de ontologías naturales. Se empieza describiendo el método de comparación Fréchet Moderno el cual es una medida de distancia elástica en dos dimensiones, posteriormente se explica la forma en la que se obtienen las ontologías naturales. Se presenta también el procedimiento para la obtención de dichas ontologías naturales, se utiliza su representación matricial para realizar la comparación.

5.1. Distancia Fréchet Moderno (FM)

Fréchet Moderno es una optimización al problema de alineación elástica en dos dimensiones (2DDW). Se realiza la comparación de matrices obteniendo la distancia mínima los datos de las matrices. Para ello se utiliza DTW que a su vez hace uso de la Distancia Euclidiana como distancia base. Fréchet Moderno extiende el método DTW a dos dimensiones, es decir, los datos no son secuencias ahora son arreglos bidimensionales, matrices numéricas, estos datos pueden ser de diferente índole como se ha mencionado anteriormente. FM se basa en la distancia Fréchet pero de tal manera que los datos se puedan analizar en R^2 .

5.1.1. Descripción

Supongamos un espacio métrico $M(D, FM)$ definido para un dominio de objetos D y una función de distancia $FM : D \times D \mapsto \mathbb{R}$. Sea $A, B \in D$ dos matrices de tamaño (P, Q) y (R, S) respectivamente, sean p, q índices para acceder a los elementos de la matriz A y sean r, s los respectivos para la matriz B . La función de distancia FM es definida por la función recursiva:

$$(p, q, r, s) = \min\{D(\text{etapas previas}) + \text{costo}(p, q, r, s)\} \quad (5.1.1)$$

$$\text{donde } \begin{cases} 1 \leq p \leq P \\ 1 \leq q \leq Q \\ 1 \leq r \leq R \\ 1 \leq s \leq S \end{cases} \quad (5.1.2)$$

Para calcular la distancia Fréchet Moderno $FM(p, q, r, s)$, se construye un arreglo de cuatro índices de tamaño (P, Q, R, S) . Se realiza un recorrido donde para cada elemento de la matriz A se hacen las operaciones sobre la matriz B , dichas operaciones están definidas en la función costo.

La función costo es definida por el cálculo de DTW utilizando secuencias $R1$ y $R2$, que son las correspondientes a los índices de entrada p, r de d , $C1$ y $C2$ utilizan los índices correspondientes q, s de entrada. Se explica con detalle más adelante.

$$\text{Costo}(p, q, r, s) = DTW(R1, R2) + DTW(C1, C2) \quad (5.1.3)$$

Las etapas previas están definidas como el número de combinaciones posibles a partir de los índices dados, esto se explica en las siguientes secciones.

5.1.2. Función costo de FM

La función costo esta definida como el alineamiento elástico entre dos secuencias o arreglos, este alineamiento es solucionado con la técnica DTW (que se mencionó en el Capítulo 3).

Sea $R1$ y $C1$ dos subsecuencias de la matriz A correspondiente a dos índices de entrada (p, q) . Sea $R1$ igual a los valores correspondientes a la fila p desde el intervalo $[1, q]$. Sea $C1$ los valores correspondientes a la columna q desde el intervalo $[1, p]$. Análogamente es para $R2$ y $C2$ utilizando los índices (r, s) . Estas secuencias son mejor observadas en el esquema de la Fig. 5.1.

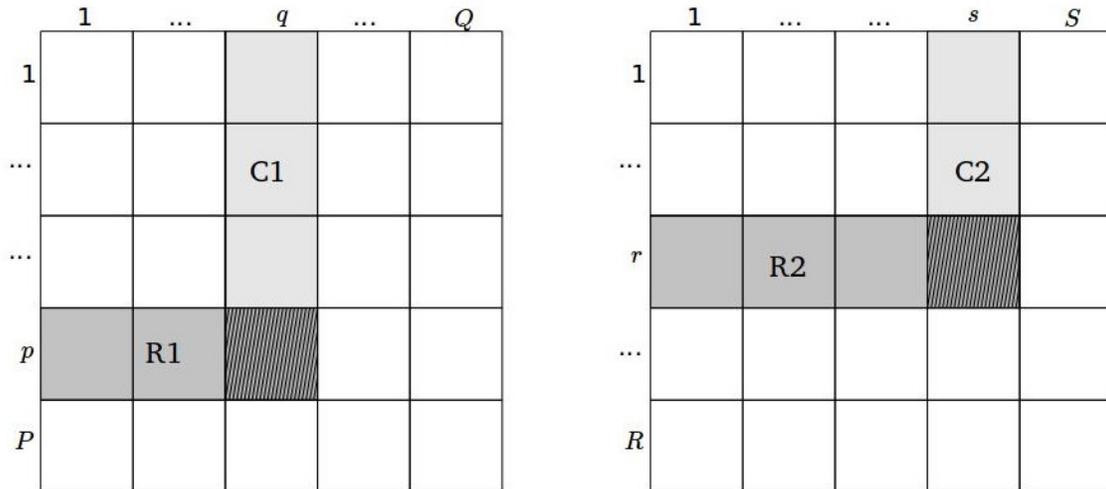


Figura 5.1: $R1$ es la fila p acotada en el intervalo $(1, q)$, $C1$ es la columna q acotada en el intervalo $(1, p)$. Similarmente para $R2$ es la fila r acotada en el intervalo $(1, s)$ y $C2$ es la columna s acotada en el intervalo $(1, r)$. Para ambas matrices A y B correspondientes.

Así, la función costo para la siguiente entrada $d(3, 4, 2, 3)$, se toman los índices $(3, 4)$ y se obtienen $R1$ y $C1$ a partir de ellos, se toman también los índices $(2, 3)$ y se obtienen $R2$ y $C2$, la función costo es aplicar $DTW(R1, R2)$ y sumarle $DTW(C1, C2)$. Conforme se van avanzando los índices de entrada, eventualmente se van realizando comparaciones de todas las posiciones de las matrices de entrada, con el objetivo

de revisar las subsecuencias y hacer una comparación elástica de ellas, el siguiente diagrama muestra como es el recorrido Figura 5.2.

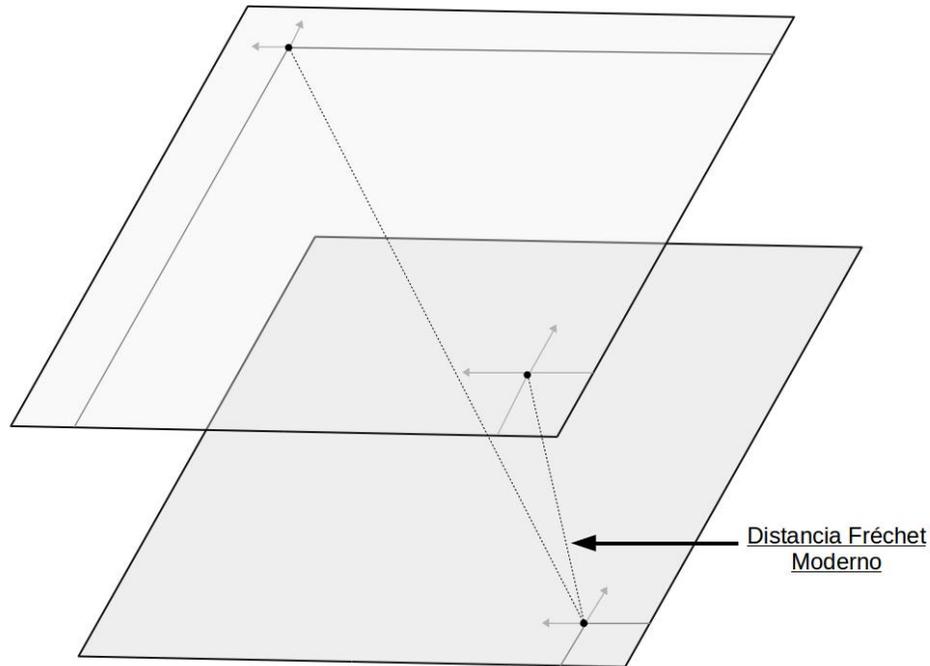


Figura 5.2: FM es un método de comparación elástica que compara la distorsión que se encuentra en filas y columnas de un elemento de la matriz A sobre todos los elementos de la matriz B , de esta manera se hace un recorrido completo en las dos matrices de entrada.

5.1.3. Etapas previas

Para calcular el valor de cada coordenada de $d(p, q, r, s)$ se tienen que calcular sus etapas previas. Para una posición de la matriz A dada por los índices (p, q) se tienen tres posibles etapas previas que están sujetas a limitaciones de fronteras:

1. $(p - 1, q - 1)$
2. $(p, q - 1)$

3. $(p - 1, q)$

Las etapas previas de $d(p, q, r, s)$ son todas las posibles combinaciones de las etapas previas de cada matriz, es decir, es la combinación de los incisos a, b, c y e, f, g de la Figura 5.3. Las combinaciones de éstas son 15, y se describen de forma completa en la Tabla 5.1.

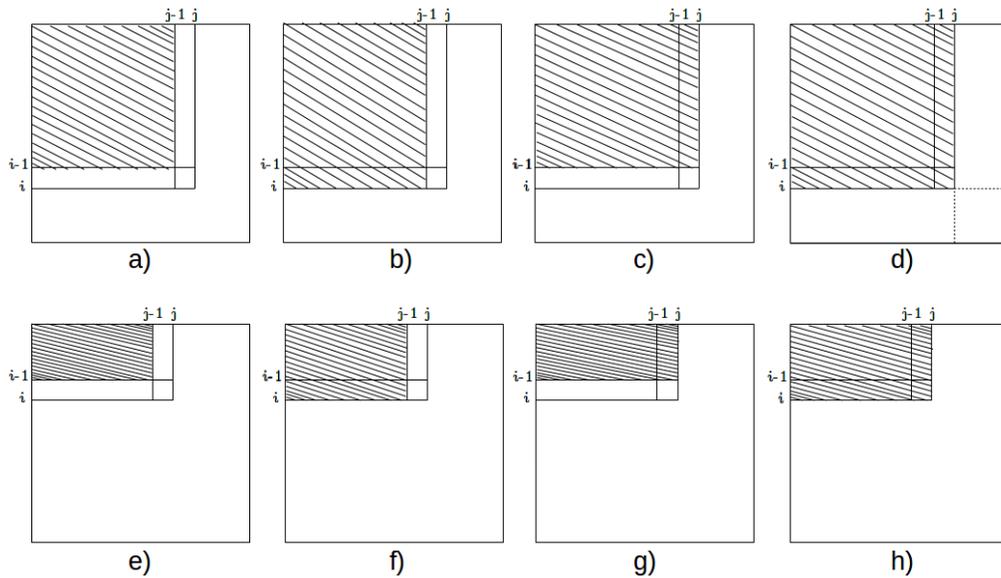


Figura 5.3: Etapas previas de la matriz A: se encuentran en el inciso a,b,c. Mientras que las etapas de la matriz B son los inciso e, f, g. Los incisos d, h son la etapas actuales de cada matriz.

Utilizando Programación Dinámica, se pueden realizar las iteraciones correspondientes para la resolución de las etapas previas con su respectivo costo, con el objetivo de reducir la complejidad computacional. Podemos generar una estructura de arreglo de cuatro índices ($d(p, q, r, s)$) que es donde se almacenan los costos mínimos acumulados durante las iteraciones. Tomando como resultado final la última posición de esta estructura, como es mostrado en el Algoritmo 1.

En R^2 se debe hacer un recorrido de alta complejidad como es mostrado en el Algoritmo 2, se muestran en pseudocódigo las 15 etapas previas, así también la

Tabla 5.1: Etapas totales con el cálculo de la función costo correspondiente a cada etapa.

#	Etapas Previa	Costo
1	$(p - 1, q - 1, r - 1, s - 1)$	$DTW(R1, R2) + DTW(C1, C2)$
2	$(p - 1, q - 1, r - 1, s)$	$DTW(R1, R2) + DTW(C1, C2)$
3	$(p - 1, q - 1, r, s - 1)$	$DTW(R1, R2) + DTW(C1, C2)$
4	$(p - 1, q - 1, r, s)$	$DTW(R1, R2) + DTW(C1, C2)$
5	$(p - 1, q, r - 1, s - 1)$	$DTW(R1, R2) + DTW(C1, C2)$
6	$(p - 1, q, r - 1, s)$	$DTW(R1, R2)$
7	$(p - 1, q, r, s - 1)$	$DTW(R1, R2) + DTW(C1, C2)$
8	$(p - 1, q, r, s)$	$DTW(R1, R2)$
9	$(p, q - 1, r - 1, s - 1)$	$DTW(R1, R2) + DTW(C1, C2)$
10	$(p - 1, q - 1, r - 1, s)$	$DTW(R1, R2) + DTW(C1, C2)$
11	$(p, q - 1, r, s - 1)$	$DTW(C1, C2)$
12	$(p, q - 1, r, s)$	$DTW(C1, C2)$
13	$(p, q, r - 1, s - 1)$	$DTW(R1, R2) + DTW(C1, C2)$
14	$(p, q, r - 1, s)$	$DTW(R1, R2)$
15	$(p, q, r, s - 1)$	$DTW(C1, C2)$

función de costo correspondiente para cada etapa. Las etapas y sus costos están definidas por una coordenada a calcular $d(p, q, r, s)$.

Algoritmo 1 Recurrencia que calcula la Distancia Fréchet Moderno

Entrada: Matrices $A(p, q)$, $B(r, s)$ de datos numéricos.

Salida: Número escalar k , indica la distancia entre A y B.

```

1:  $P \leftarrow$  Filas de  $A$ 
2:  $Q \leftarrow$  Columnas de  $A$ 
3:  $R \leftarrow$  Filas de  $B$ 
4:  $S \leftarrow$  Columnas de  $B$ 
5:  $d(P, Q, R, S)$  Matriz de 4 índices, inicializada en  $\infty$ 
6: para  $p = 2$  hasta  $P$  hacer
7:   para  $q = 2$  hasta  $Q$  hacer
8:     para  $r = 2$  hasta  $R$  hacer
9:       para  $s = 2$  hasta  $S$  hacer
10:          $d(p, q, r, s) = \min\{ \text{costo}(p, q, r, s) + \text{EtapasPrevias}(p, q, r, s) \}$ 
11:       fin para
12:     fin para
13:   fin para
14: fin para
15: devolver  $k = d(P, Q, R, S)$ 

```

5.2. Elicitación de Ontologías Naturales

La anotación de ontologías se refiere al proceso de creación de éstas mediante el uso de Redes Semánticas Naturales, en el Capítulo 2 se mencionó que éstas son utilizadas para el proceso de anotación, las RSN se encuentra en los estudios de la memoria semántica y en las redes semánticas.

A grandes rasgos, el procedimiento es aplicado a un grupo de personas, se les proporciona un cuestionario individual para que definan conceptos de un cierto dominio, a partir de esos cuestionarios se anotan las Ontologías Naturales.

Algoritmo 2 Algoritmo para calcular la distancia Fréchet Moderno**Entrada:** Matrices $A(p, q)$, $B(r, s)$ de datos numéricos.**Salida:** Número escalar k , indica la distancia entre A y B.

```

1:  $P \leftarrow$  Filas de  $A$ 
2:  $Q \leftarrow$  Columnas de  $A$ 
3:  $R \leftarrow$  Filas de  $B$ 
4:  $S \leftarrow$  Columnas de  $B$ 
5:  $d(P, Q, R, S)$  Matriz de 4 índices, inicializada en  $\infty$ 
6: para  $p = 2$  hasta  $P$  hacer
7:   para  $q = 2$  hasta  $Q$  hacer
8:     para  $r = 2$  hasta  $R$  hacer
9:       para  $s = 2$  hasta  $S$  hacer
10:          $d(p, q, r, s) = \min \left\{ \begin{array}{l}
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q-1, r-1, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q-1, r-1, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q-1, r, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q-1, r, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q, r-1, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q, r-1, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q, r, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p-1, q, r, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q-1, r-1, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q-1, r-1, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q-1, r, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q-1, r, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q, r-1, s-1) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q, r-1, s) \\
DTW(R1, R2) + DTW(C1, C2) + d(p, q, r, s-1)
\end{array} \right\}
11:       fin para
12:     fin para
13:   fin para
14: fin para
15: devolver  $k = d(P, Q, R, S)$$ 
```

El proceso consta de seis pasos que se pueden observar de manera global en la Fig. 5.4. Una característica importante acerca de esta proceso es que puede ser aplicado en cualquier dominio para obtener ontologías naturales, en lo que respecta a este trabajo, se ha utilizado para obtenerlas en el dominio de las Ciencias de la

Computación, en el Apéndice C, se muestra cómo se realizó el procedimiento para obtener la forma matricial de las ontologías para poder hacer uso de la técnica de comparación Fréchet Moderno, que también se explica en el Capítulo 6.

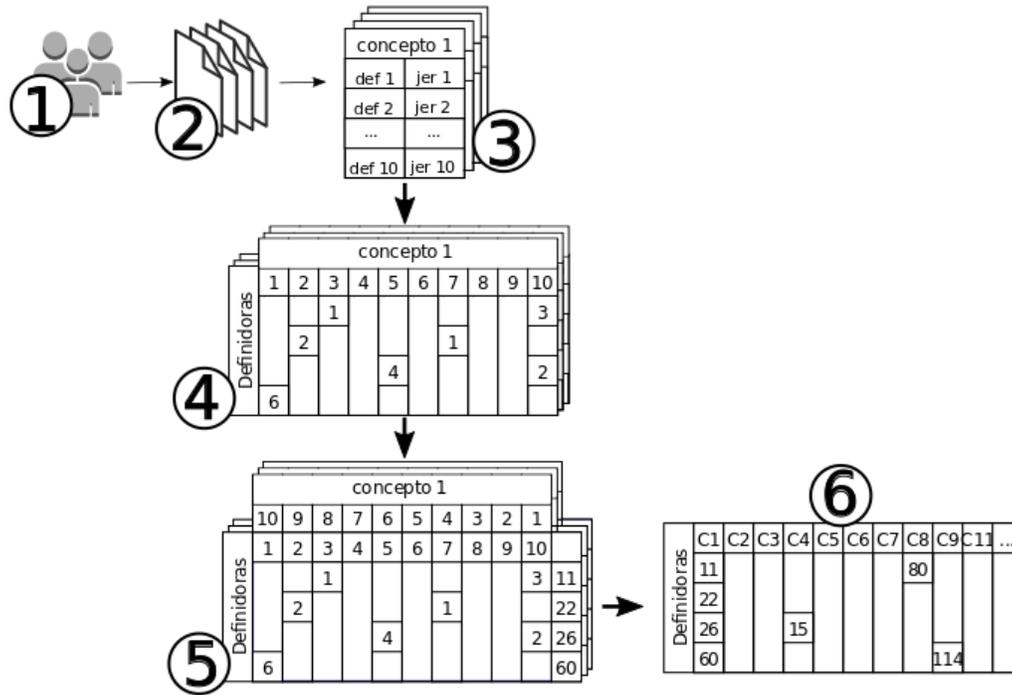


Figura 5.4: Esquema del proceso de construcción de RSN.

5.2.1. Técnica de Redes Semánticas Naturales

Se describe la metodología para obtener ontologías naturales a partir de protocolos contestados por sujetos dentro de un dominio de conocimiento, la metodología es propuesta por [Figuroa *et al.*, 1976], ha sido muy utilizada para hacer estudios dentro de diferentes dominios del conocimiento, con resultados favorables [García-Cabrero y Jiménez-Vidal, 1996, Figuroa y Carrasco, 1980, Cabalín-Silva y Navarro-Hernández, 2008, Petra-micu *et al.*, 2012].

PASO 1 y 2 - Grupo de personas y Protocolo

El protocolo se refiere a un cuestionario en forma encuesta, el cual debe ser contestado por un grupo de 30 o más personas. Estos sujetos pertenecen a un mismo dominio de conocimiento (biología, medicina, computación, etc.). Es suficiente 30 sujetos para determinar el conocimiento de un grupo [Valdez-Medina, 2004].

El contenido del protocolo se divide en dos actividades:

1. Definir con la mayor precisión posible a un Concepto-estímulo, mediante la utilización de un mínimo de 5 conceptos definidores, que pueden ser verbos, adverbios, sustantivos, adjetivos, nombres, pronombre, sin utilizar artículos o preposiciones de algún tipo. Hay estudios [Figueroa *et al.*, 1981] que muestran que el conjunto mínimo de definidores es de 5, entre mayor sea el número de éstas es mejor, debido a que aporta un valor semántico mayor.
2. Ya que se haya definido el Concepto-estímulo, se le solicita al sujeto que, jerarquice todos los conceptos definidores que proporcionó, en función a la importancia que considere que tiene cada una de los definidores con respecto al estímulo definido. De esta manera le asignaran el número 1 al concepto definidor más cercano al concepto-estimulo, el 2 al definidor que le sigue en importancia y así sucesivamente hasta completar todos las definidores.

De esta manera, por cada grupo de dominio que se requiera, se tendrán 30 protocolos. La forma de elegir los conceptos-estímulo es una selección cuidadosa, en la cual no se prioriza el hecho de no escoger conceptos aislados, sino aquellos que puedan incluso formar parte de los definidores dados, con el objetivo de crear una red [Valdez-Medina, 2004]. Cada concepto cuenta con espacio para 15 definidores pero el sujeto puede dar entre 10-15 definidores por concepto.

En este caso, se realizó un protocolo con 18 conceptos-estímulo del dominio de Computación, siendo los conceptos de la Tabla 5.2, en el Apéndice A, se muestra el

formato del protocolo utilizado para este trabajo.

Tabla 5.2: *Conceptos usados para el protocolo del dominio de conocimiento de computación*

ESTRUCTURA	COMPLEJIDAD	CPU
MEMORIA	ALGORITMO	ALU
RED	LENGUAJE	USB
S.O.	APUNTADOR	KERNEL
INFORMACIÓN	OPTIMIZAR	URL
PROCESO	GPU	BIOS

Con los protocolos contestados se procede a la obtención de 4 valores principales que ayudan al análisis de las redes semánticas naturales, se utilizarán a lo largo de las siguientes secciones, estos son:

- **Valor J:** Este valor se refiere al número total de definidores que fueron generados por los sujetos para definir un estímulo, es un indicador de la riqueza semántica de la red, de tal manera que entre mayor sea el número de definidoras obtenidas, mayor sera la riqueza de la red.
- **Valor M:** Es la multiplicación de la **frecuencia** de aparición de la jerarquía obtenida por el peso semántico de esa jerarquía, por ejemplo, el concepto-estímulo R ha tenido 10 definidoras, el definidor F ha sido jerarquizado con el número 2 por 5 personas del grupo, entonces su frecuencia de aparición es 5. Como se tuvieron 10 definidores para este concepto-estímulo, entonces el peso semántico de la jerarquía 2 es 9, conforme a la Tabla 5.3 el definidor F tuvo una frecuencia de aparición de 2 y un peso semántico por lo tanto el valor M del definidor F es de $5 * 9 = 45$.
- **Valor M total (VMT):** El Valor M Total de un concepto definidor es la

Tabla 5.3: Pesos semánticos para el número de definidores obtenidos.

Concepto-Estímulo										
Peso Semántico	10	9	8	7	6	5	4	3	2	1
Jerarquía	1	2	3	4	5	6	7	8	9	10
Definidor F	•	5*9	•	•	•	•	•	•	•	•

suma de todas las frecuencias de aparición que tenga multiplicadas con sus respectivos pesos semánticos.

- **Conjunto SAM:** Es un grupo de conceptos definidores con mayor número M total, es decir, las n palabras con mayor VMT, el tamaño del conjunto puede variar, se toman más de 15 palabras debido a que ese número es mas representativo del significado de un concepto [Valdez-Medina, 2004].
- **Valor FMG:** Valor que se obtiene para todos los conceptos que forman SAM, obteniendo el porcentaje de la distancia semántica de los definidores incluidos en SAM; se obtiene realizando una regla de tres, tomando como punto de partida la palabra definidora con VMT más alto, representará el 100 %.



La metodología de obtención de las Redes Semánticas Naturales no es una metodología estadística, sino que usa medidas frecuentistas y no utiliza promedios que de alguna forma suponen datos acomodados de manera gaussiana o alguna otra distribución. Por lo tanto se toman los valores “crudos” de la frecuencia de aparición de los conceptos definidores.

Las medidas obtenidas de las RSN (Valor J, Valor M, VMT, SAM, FMG) son medidas cuantitativas frecuentistas. Estas son una forma de distancia relativa dada por los datos de repetición.

La Elicitación de conocimiento mediante el uso de redes semánticas naturales originalmente utiliza $SAM = 15$ para limitar la cantidad de definidoras que se toman en cuenta. A diferencia del trabajo original [Figuroa, 1981], este trabajo contempla el uso de más definidoras, debido a dos razones importantes, 1) el poder de computo no es una limitante como lo fue en su tiempo cuando surgió el trabajo de Redes Semánticas Naturales, y 2) porque al considerar más definidoras se enriquece la red semántica, porque se analizan todas las relaciones que se pueden formar entre todas la definidoras.

PASO 3 - Captura de Información

Una vez se tengan los protocolos contestados, se deben vaciar en computadora, para esto se pueden utilizar hojas de cálculo en Excel. Se recomienda que este proceso de vaciado sea realizado a mano por una persona debido a que así se corregirán algunos errores de ortografía y semántica.

De cada concepto-estímulo se obtiene una lista de los conceptos definidores con su correspondiente jerarquía. Como es mostrado en la Tabla 5.4, donde en la primera columna se tiene el definidor y en la segunda columna se encuentra su jerarquía.

Listar los definidores que todos los sujetos proporcionaron al primer concepto-estímulo, después se hace lo mismo para el segundo concepto-estímulo y así sucesivamente hasta completar todos los conceptos-estímulo. Ésta lista de definidores que se tiene por cada concepto-estímulo es variable y cae en un rango de 100 – 200 definidores.

PASO 4 - Conteo de definidores

Después se toma la lista completa de definidores de un solo concepto y se hace una tabla. Esta nueva tabla tiene el siguiente formato: la lista de definidores que dieron todos los sujetos sobre el mismo concepto se encuentra en la primera columna,

Tabla 5.4: Lista de definidores de un grupo para un Concepto-estímulo.

Concepto-estímulo	
Definidor 1	3
Definidor 2	1
Definidor 3	4
Definidor 4	7
Definidor 5	2
Definidor 6	8
Definidor 7	10
Definidor 8	6
...	...
Definidor m	m

como se muestra en la Tabla 5.5. En la última columna de esta tabla se tiene el VMT de cada definidor.

Tabla 5.5: Formato de la tabla de captura para el vaciado de los cuestionarios.

ESTRUCTURA											
Peso Semántico	10	9	8	7	6	5	4	3	2	1	
Jerarquía	1	2	3	4	5	6	7	8	9	10	VMT
Definidor 1	•	•	•	•	•	•	•	•	•	•	
Definidor 2	•	•	•	•	•	•	•	•	•	•	
...	•	•	•	•	•	•	•	•	•	•	
Definidor n	•	•	•	•	•	•	•	•	•	•	

La primera fila de la tabla contienen una numeración del 1 al 10, indicando la jerarquía posible que los sujetos puedan otorgar a un definidor en particular, así

se coloca un “/” en la columna del número de la jerarquía que le asignaron a un definidor.

Veamos esto con un ejemplo sencillo, en la Tabla 5.6 se muestran protocolos de muestra que sirven para ilustrar el proceso. En el caso del concepto definidor “Datos”, el sujeto 1 le asigna la jerarquía 1 al definidor Datos, mientras que el sujeto 2 le asigna una jerarquía 2 y el sujeto 3 la jerarquía 2, para el sujeto 2 y 3, la jerarquía otorgada fue la misma, por lo tanto debe ser acumulada.

Es muy probable que existan definidores que no se repitan entre sujetos, en ese caso solo se coloca un 1 que indica la frecuencia de aparición fue de una sola persona, sin embargo, estos definidores también forman parte de la riqueza semántica y son sumadas en el VMT.

Tabla 5.6: Tabla que captura las ocurrencias de los definidores

Sujeto 1		Sujeto 2		Sujeto 3	
Definidores	Jerarquía	Definidores	Jerarquía	Definidores	Jerarquía
Datos	1	Programa	5	Datos	2
Información	3	Árbol	1	Edificio	1
Edificio	4	Cadena	3	Información	5
Metal	2	Datos	2	Árbol	3
Programa	5	Información	4	Metal	4
Árbol	8			Matriz	6
Arreglo	7			Arreglo	7
Matriz	6				

La captura de esta información tiene la siguiente organización (ver Tabla 5.7), cada “/” significa que un sujeto utilizó el concepto definidor con una jerarquía específica. Esto es un paso intermedio, para que al final esas “/” se conviertan en

números, así la información de las repeticiones queda como en la Tabla 5.8.

Tabla 5.7: Tabla que captura las ocurrencias de los definidores

ESTRUCTURA											
Valor Semántico	10	9	8	7	6	5	4	3	2	1	
Jerarquía	1	2	3	4	5	6	7	8	9	10	VMT
Información			/	/	/						
Datos	/	//									
Edificio	/			/							
Metal		/		/							
Árbol	/		/					/			
Arreglo						/	/				
Matriz						/	/				
Programa					//						
Cadena			/								

PASO 5 - Pesos Semánticos

Una vez realizado este procedimiento, se obtiene el valor M de cada definidora, por lo tanto, se realiza el cálculo del Valor M de todas las definidora. El VMT es la suma de esos valores, que se calcula en la última columna de nuestra tabla 5.9.

Obtenemos el **peso semántico** de cada ocurrencia en la tabla anterior, esto se hace multiplicando la frecuencia de repetición de un definidor por el valor semántico correspondiente, este valor semántico se encuentra en la segunda fila de la tabla. Si existe más de una ocurrencia para un definidor, se suma. Nota: si se tienen más de 10 definidoras por concepto, entonces la jerarquía aumenta, en vez de ser 10 será el número de conceptos definidores que se hayan registrado.

Tabla 5.8: Tabla que captura las ocurrencias de los definidores, convertidas en números.

ESTRUCTURA											
Valor Semántico	10	9	8	7	6	5	4	3	2	1	
Jerarquía	1	2	3	4	5	6	7	8	9	1	VMT
Información			1	1	1						
Datos	1	2									
Edificio	1			1							
Metal		1		1							
Árbol	1		1					1			
Arreglo						1	1				
Matriz						1	1				
Programa					2						
Cadena			1								

Cabe aclarar que en este ejemplo, la jerarquización va de 1–10, sin embargo, puede variar el rango dependiendo de los sujetos, si un solo sujeto escribió 15 definidoras para algún concepto, este tuvo que haber jerarquizado las definidoras del 1 al 15, por lo tanto la tabla debe crecer en ese rango, de 1 – 15.

Con el procedimiento anterior se logran los pesos semánticos para las definidoras de un concepto, lo que falta por mencionar, es que este proceso debe repetirse para todas los conceptos-estímulo que se presentan en el protocolo, esto implica que se deben crear tablas como la Tabla 5.9.

PASO 6 - Obtención de la Ontología Natural

El paso siguiente es obtener una matriz global por grupo, es decir, se unen los definidores de todos los conceptos-estímulo en una sola tabla. Para crearla se utilizan los conceptos-estímulo, los definidores de estos y el VMT de cada definidor. De igual

Tabla 5.9: Tabla que captura las ocurrencias de las definidoras con valores semánticos.

ESTRUCTURA											
Valor Semántico	10	9	8	7	6	5	4	3	2	1	
Jerarquía	1	2	3	4	5	6	7	8	9	1	VMT
Información			$1 * 8 = 8$	$1 * 7 = 7$	$1 * 6 = 6$						$8 + 7 + 6 = 21$
Datos	$1 * 10 = 10$	$2 * 9 = 18$									$10 + 18 = 28$
Edificio	$1 * 10 = 10$			$1 * 7 = 7$							$10 + 7 = 17$
Metal		$1 * 9 = 9$		$1 * 7 = 7$							$9 + 7 = 16$
Árbol	$1 * 10 = 10$		$1 * 8 = 8$					$1 * 2 = 2$			$10 + 8 + 2 = 20$
Arreglo						$1 * 5 = 5$	$1 * 4 = 4$				$5 + 4 = 9$
Matriz						$1 * 5 = 5$	$1 * 4 = 4$				$5 + 4 = 9$
Programa					$2 * 6 = 6$						6
Cadena			$1 * 8 = 8$								8

manera, como se realizó para cada concepto, se tienen que encontrar las ocurrencias del definidor a través de todos los estímulos y sumarla en caso de ocurrencia, creando así la matriz global. Un ejemplo es mostrado en la Tabla 5.10.

Tabla 5.10: Tabla global de valores semánticos para los conceptos-estímulo

•	Estructura	Memoria	Red	S.O.	Información	Proceso	Complejidad	Algoritmo	Lenguaje	Apuntador	Optimizar	GPU	CPU	ALU	USB	Kernel	URL	BIOS
Definidora 1	89	•	•	•	•	23	•	•	•	33	•	123	•	22	123	47	•	55
Definidora 2	•	65	•	54	•	65	42	•	85	•	22	•	33	65	•	•	55	5
...	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Definidora m	•	•	32	•	12	•	90	•	123	•	13	111	11	•	77	•	99	4

Si se realiza una gráfica de la matriz resultante se obtienen gráficas (Fig. 5.5), donde por un lado se encuentran los conceptos definidos y por el otro los concepto definidores, dentro de la matriz se encuentran los pesos semánticos de cada definidor.

El peso semántico es un valor acumulativo de las ocurrencias de una definidora, por ejemplo en la Fig. 5.5 la definidora 80 tiene un peso semántico de aproximadamente de 150, mientras que la definidora 82 tiene un peso semántico mayor o igual a 300.

El código de colores indica que entre más grande es el valor de una definidora, se aproxima más al color rojo.

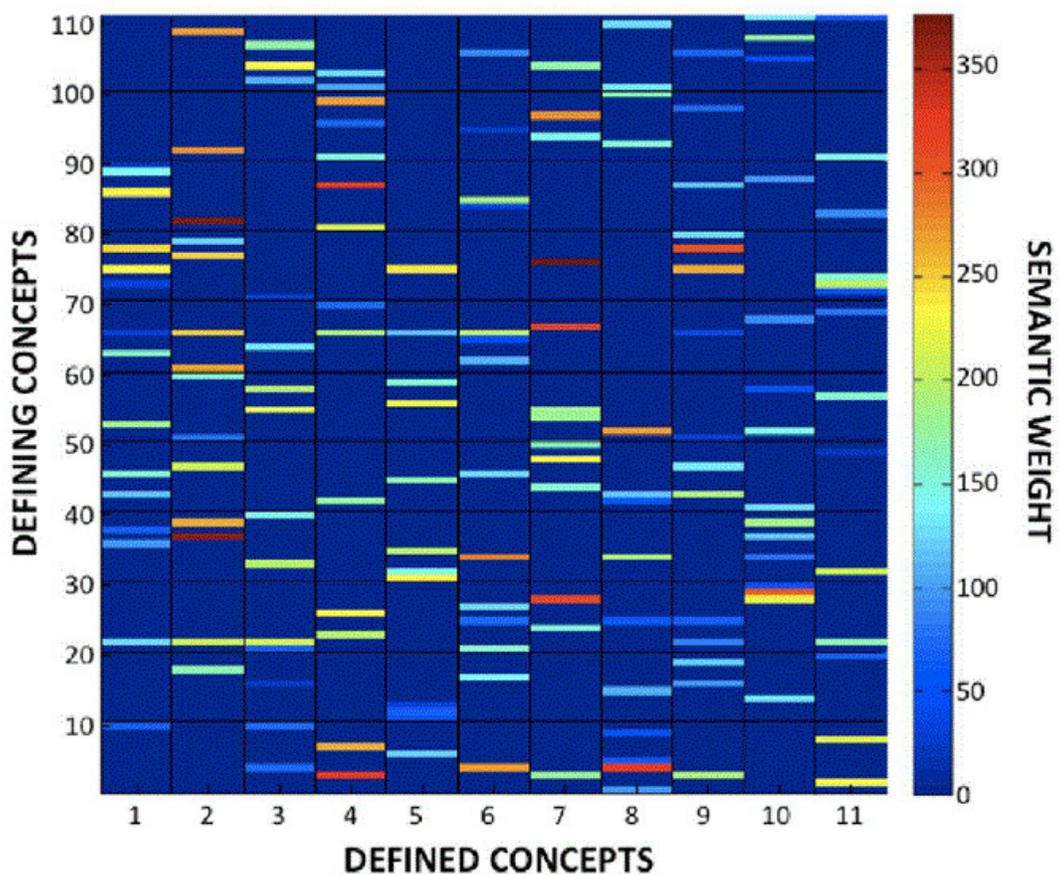


Figura 5.5: Representación matricial de una ontología

Capítulo 6

Resultados

En este capítulo se realiza la obtención de conocimiento de grupos de personas y es comparado utilizando la técnica de distancia Fréchet Moderno. También se muestran experimentos utilizando imágenes y matrices artificiales.

6.1. Comparación de Ontologías Naturales

La comparación de ontologías naturales es realizada utilizando la distancia de Fréchet Moderno que toma dos matrices de datos numéricos, éstas matrices son las que se obtienen del proceso de elicitación de ontologías naturales. Se hace uso del índice SAM para poder comparar, éste índice es compuesto por las definidoras que tienen mayor VMT. Se escogen las 15, 20, 30, 40 ó 50 definidoras con mayor VMT. En este trabajo se ha escogido que el SAM sea de 20.

La elicitación de conocimiento mediante el uso de la técnica de ontologías naturales se ha realizado a tres grupos de personas en el dominio de computación, se tienen 3 grupos de 20 personas a las que se les aplicó el cuestionario y a partir de éste se realizó el proceso de la creación de ontologías.

Los grupos fueron:

1. Estudiantes de posgrado del Centro de Investigación en Computación (CIC).
2. Estudiantes de 8vo semestre de la Escuela Superior de Computación (ESCOM).
3. Estudiantes de 8vo semestre en Computación del Tecnológico de Iztapalapa (IZTA).

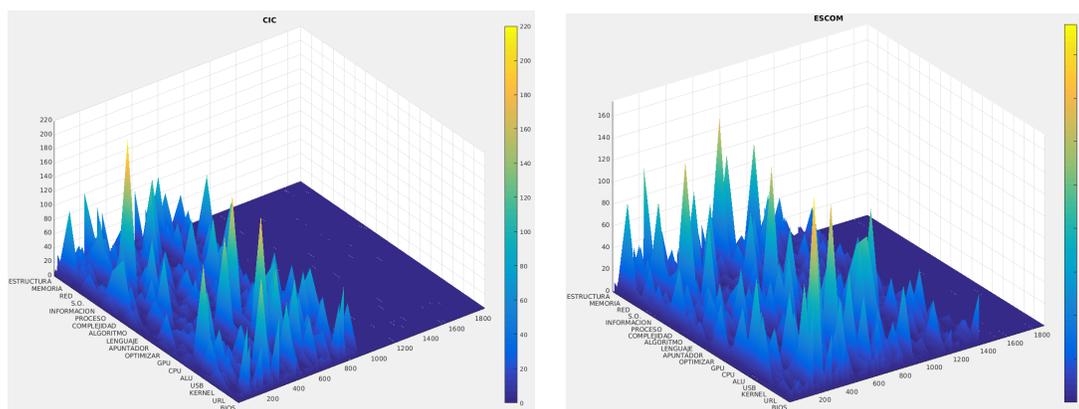
El grupo de estudiantes de posgrado se ha tomado como el referente con mayor nivel de estudios por lo tanto es considerado como el grupo experto, las comparaciones se realizarán en torno a esta pre-suposición. Se han obtenido tres ontologías naturales, una por cada grupo, éstas se han sometido a la comparación utilizando la técnica Fréchet Moderno. Antes de ser comparadas es necesario realizar un proceso de unión.

Proceso de unión: el vocabulario de definidoras de cada grupo es diferente, se pueden dar los siguientes casos:

1. Las definidoras que estén en un grupo pero que no estén en los otros dos grupos.
2. Las definidoras que se encuentran en dos grupos y en uno no.
3. Las definidoras que sean comunes en los tres grupos.

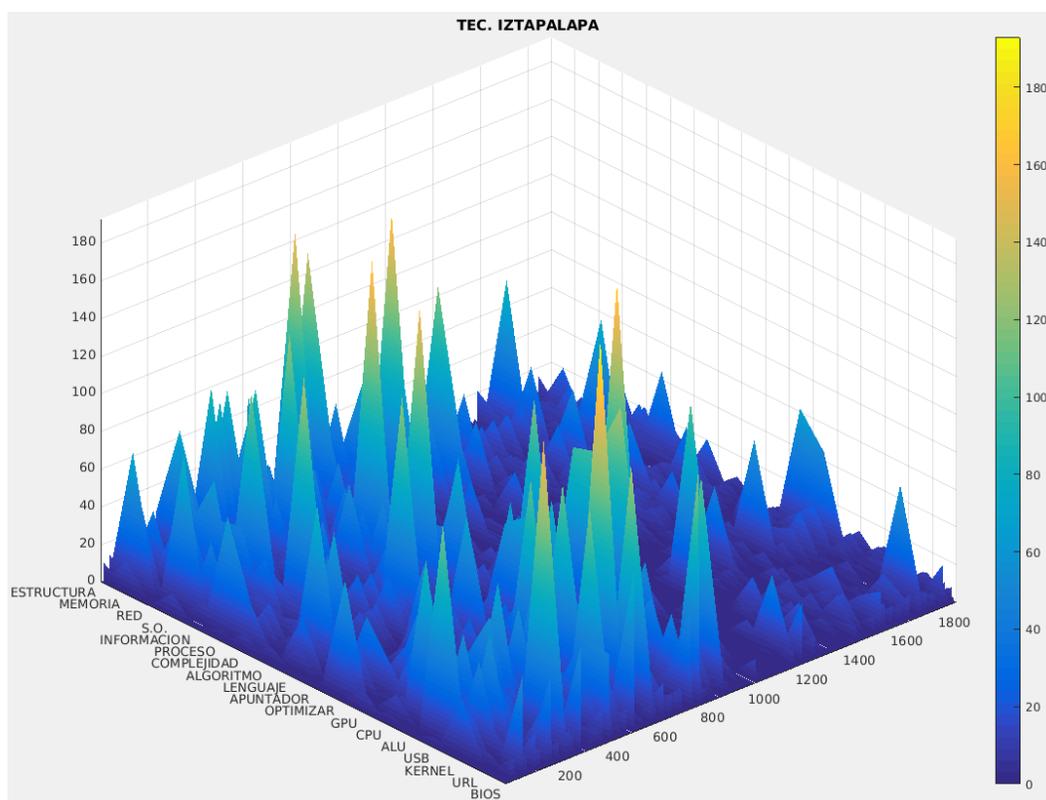
El proceso de unión consiste en fusionar todas las definidoras de todos los grupos en una matriz global, se mantienen los pesos semánticos de los grupos por separado. Por lo tanto, ésta matriz global unifica la posición de las definidoras, pero mantiene los pesos semánticos de cada grupo. De esta manera se tienen tres matrices con todas las definidoras pero lo que cambia son los pesos semánticos, como las matrices de la Tabla 6.1.

Teniendo las matrices unificadas después del proceso de unión, es posible compararlas porque ahora cada una tiene la misma posición de sus definidoras y de sus conceptos. Podríamos obtener su distancia entre ellas utilizando Fréchet Moderno; sin embargo, estas matrices son demasiado grandes para la técnica de distancia, debido al recorrido complejo que realiza, su cálculo es complejo computacionalmente.



CIC

ESCOM



TEC. IZTAPALAPA

Tabla 6.1: Unión de definidoras. El eje x corresponde a los conceptos definidos, el eje y son las definidoras de los 3 grupos, eje z pesos semánticos

Es por esto que para la comparación es necesario obtener el grupo SAM de cada matriz. El grupo SAM, son todas las n definidoras con mayor peso semántico dentro del grupo, las 20, 30 o 40 definidoras.

Una vez obtenidas las matrices SAM de cada grupo, se realiza el mismo proceso de unión para estas nuevas matrices, para que queden alineadas. Así, se tienen matrices más pequeñas que corresponden al vocabulario más frecuente de cada grupo. Las matrices de cada grupo se pueden observar en la Tabla 6.3.

Una vez obtenidas las matrices de los 3 grupos SAM, podremos compararlas entre ellas utilizando la distancia Fréchet Moderno, hacemos la matriz de distancias entre grupos (Tabla 6.2), se puede analizar que la distancia en conocimiento del grupo CIC contra ESCOM es más chica que la distancia que hay entre CIC y el Tec. de Iztapalapa.

Tabla 6.2: Distancias entre grupos.

	CIC	ESCOM	TEC. IZTAPALAPA
CIC	0	910	1244
ESCOM	910	0	1087
TEC. IZTAPALAPA	1244	1087	0

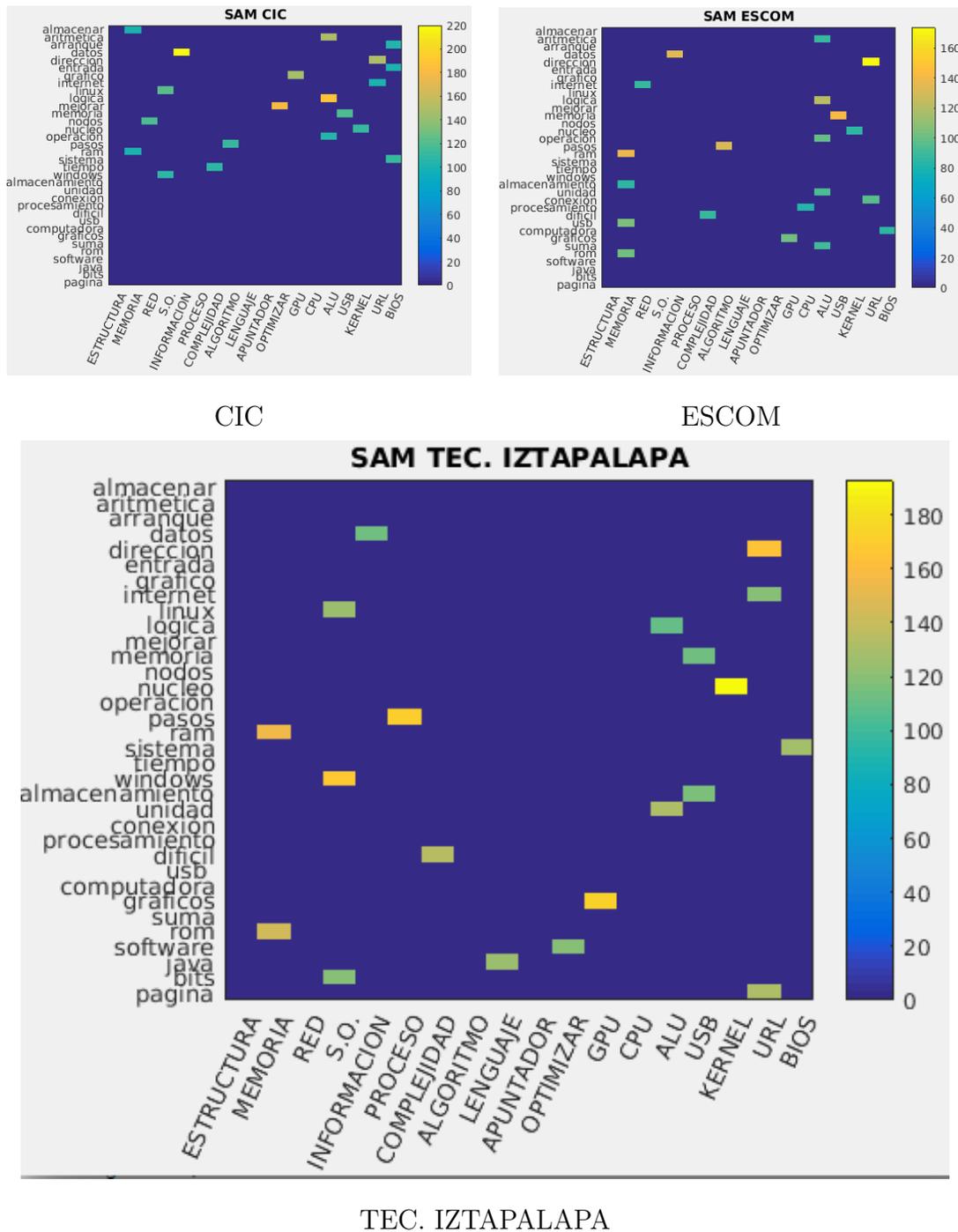


Tabla 6.3: Unión de definidoras en las matrices SAM

6.2. Comparación de Imágenes

Para realizar pruebas experimentales utilizando imágenes se ha considerado el conjunto de imágenes de Princeton, se encuentra organizada en categorías de imágenes, como se muestra en la Figura 6.1. Las imágenes del primer renglón corresponden a la categoría de los autobuses, mientras que las imágenes de la cuarta categoría son dinosaurios.

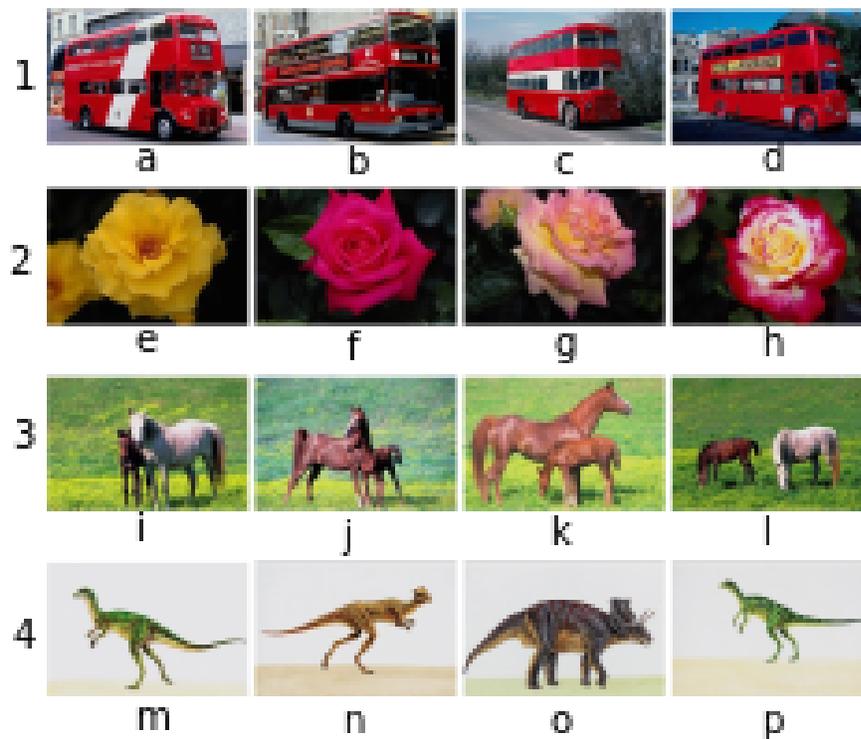


Figura 6.1: Muestra de las imágenes dentro de la base de datos de imágenes de Princeton.

El experimento consiste en realizar una búsqueda de un elemento *query* en el conjunto de elementos, de esta manera la técnica Fréchet Moderno debe mostrar cuales son los elementos que están más cercanos al elemento escogido como *query*.

Para realizar esto se ha realizado un preprocesamiento al conjunto de imágenes, este proceso incluye un reescalamiento del tamaño de la imagen para reducir su

tamaño, se considera un sólo canal de RGB de la imágenes, para esto las imágenes quedan en blanco y negro.

Para la ejecución de este experimento la imagen *query* fue la marcada con la letra *m* en la Figura 6.1, por lo tanto la lista de las imágenes mas cercanas a ella son las mostradas en la Figura 6.2, de esto se puede observar que las imágenes más cercanas al *query* son las imágenes que pertenecen a la misma categoría.

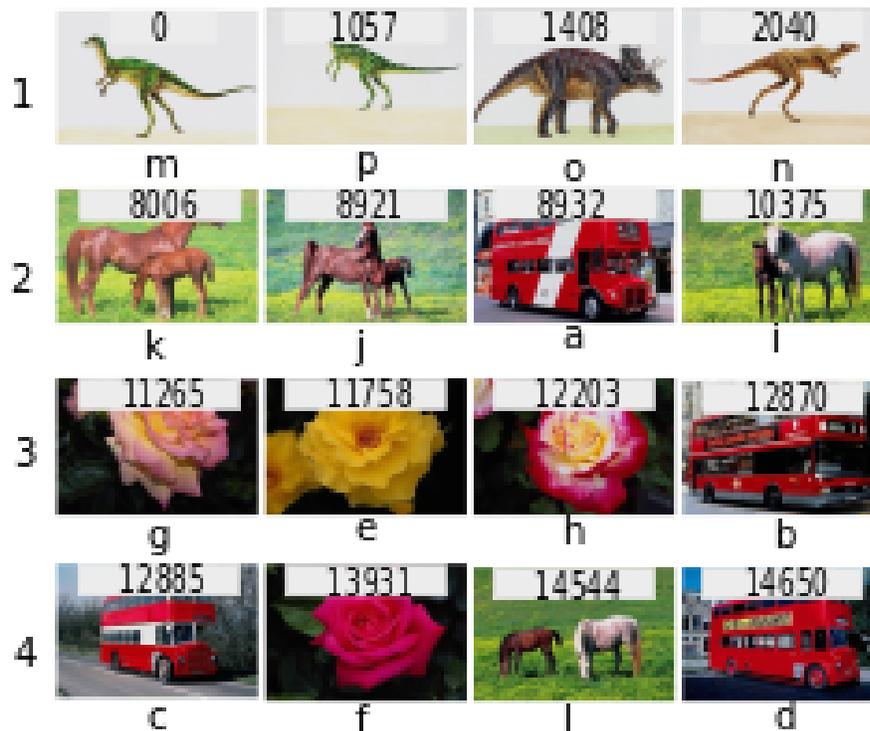


Figura 6.2: Resultados de la técnica Fréchet Moderno utilizando imágenes.

6.2.1. Validación con imágenes

En el Apéndice E se presenta un script de Matlab que realizar una validación de la técnica de distancia Fréchet Moderno.

El experimento consiste en realizar un análisis de una tarea de recuperación sobre un conjunto de datos, utilizando diferentes técnicas de comparación matricial. Se

ejecuta la tarea buscando un *query* en diferentes categorías de los objetos analizados, así se puede sacar información estadística para saber en qué medida es funcional la técnica Fréchet Moderno.

El script controla la tecnica de distancia mediante una variable `indexTecnica` tiene un rango de 1-10:

Tabla 6.4: Técnicas de Comparación Matricial

Técnicas		
Euclidiana	q	ERGAS
Bias	RASE	DIV
Coefficiente de Correlación	RMSE	Hausdorff
Información Mutua	Fréchet Moderno	

La tarea de recuperación es sencilla, debido a que la técnica Fréchet Moderno es de alta complejidad y lleva largo tiempo en hacer cálculos exhaustivos.

Se tienen un directorio de imágenes que pertenecen a diferentes categorías, ese directorio es `IMAGENES/Busqueda`, contiene 5 imágenes que son de 5 categorías, por lo tanto, se realiza una búsqueda en 25 imágenes.

Se realiza la comparación de todas las imágenes contra todas dentro de el directorio dicho, realizando una matriz de distancias entre los 25 elementos. Calculando la distancia Euclidiana, la distancia de Información Mutua y Fréchet Moderno.

Posteriormente para cada columna de la matriz de distancias j , que representa cada elemento, se verifica cuantos de los primeros 5 elementos de cada columna pertenecen a la categoría del elemento j . Se realiza un promedio de las pertenencias por cada elemento.

El análisis comienza en ejecutar el script `dendograma.m`, inicializará las imágenes y hará las comparaciones para crear la matriz de distancias, utilizando una técnica en específico: distancia euclidiana.

El resultado de las pertenencias obtenidas es:

$$\begin{array}{l}
 1 \text{ EUCLI} = [(5/5), (5/5), (5/5), (5/5), (4/5), \dots \\
 2 \quad \quad (2/5), (2/5), (1/5), (2/5), (1/5), \dots \\
 3 \quad \quad (1/5), (2/5), (2/5), (2/5), (3/5), \dots \\
 4 \quad \quad (5/5), (5/5), (5/5), (5/5), (5/5), \dots \\
 5 \quad \quad (4/5), (4/5), (4/5), (4/5), (3/5)]; \\
 6 \quad \text{EUCLI} = \text{sum}(\text{EUCLI})/25 \\
 7 \quad \quad = 0.6880
 \end{array}$$

Para la columna 1 se tuvo una recuperación de $5/5$, es decir la distancia euclidiana recupero 5 elementos de 5 para la categoría del elemento $j = 1$ de la matriz de distancias, así sucesivamente hasta completar todas las columnas.

Así el índice de recuperación de las siguientes técnicas se muestra en la Tabla 6.5.

Tabla 6.5: Comparación de los índices de recuperación de diferentes técnicas

Técnica	índice de recuperación
Euclidiana	0.6880
RASE	0.6640
Hausdorff	0.5120
Fréchet Moderno	0.6960

6.3. Comparación de Funciones Matemáticas

De manera similar, se han comparado matrices de datos artificiales, estos datos han sido de dos tipos, datos generados manualmente y datos generados con funciones matemáticas, para ambos casos se realiza el experimento de búsqueda.

6.3.1. Datos Artificiales I

Los datos generados manualmente consisten en generar una ventana de números que se van moviendo a lo largo de la matriz y así se generan diferentes matrices con ventanas de números en diferentes posiciones, la Figura 6.3 muestra la organización de la ventana que se mueve a lo largo de la matriz, por lo tanto se obtienen nueve matrices con la ventana en alguna posición. En el Apéndice F se muestra el script en Matlab que realiza la ventana y genera las matrices deseadas.

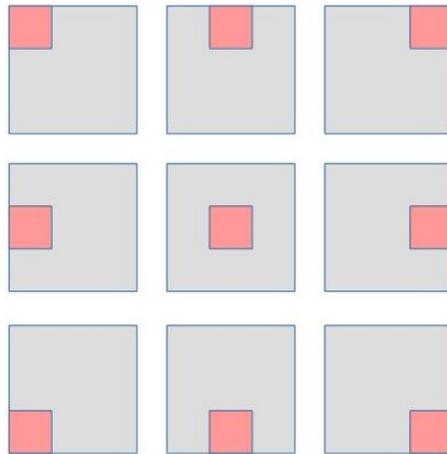


Figura 6.3: Organización de la ventana de datos.

Estas matrices se sometieron a la comparación realizada en la técnica Fréchet Moderno obteniendo como resultado las matrices de la Figura 6.4. Se ha tenido el mismo comportamiento para este tipo de datos artificiales de diferente tamaño, se realizaron pruebas con matrices cuadradas de tamaño 5,7,18 y 25.

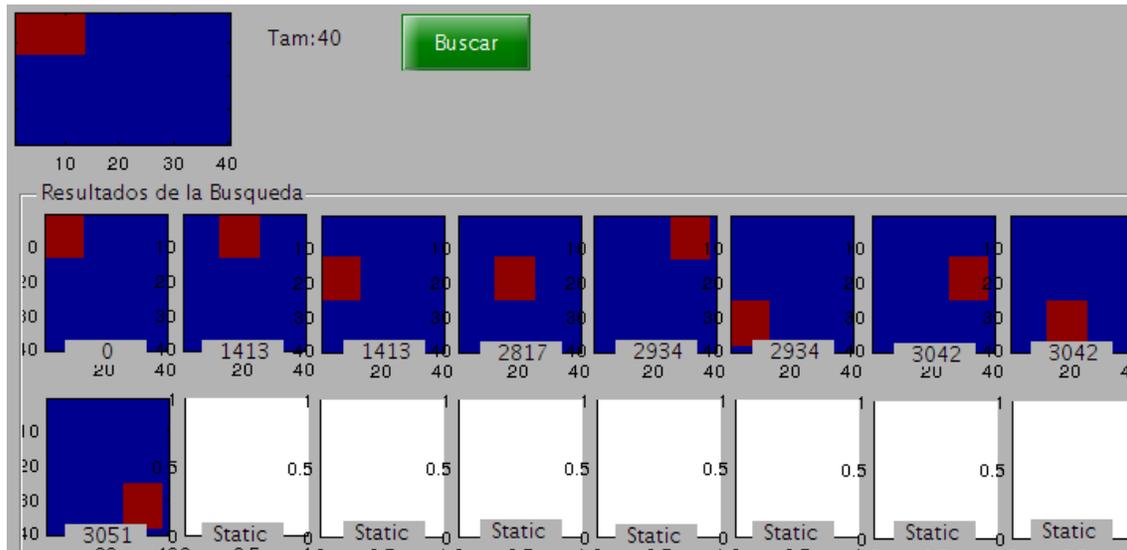


Figura 6.4: Resultados del experimento de matrices artificiales moviendo ventana de datos

6.3.2. Datos Artificiales II

Otros datos generados de manera artificial son los creados a partir de funciones matemáticas, se tienen un conjunto de funciones definidas que toman valores en el eje x y en eje y y calculan valores para el eje z , así los valores de la función se encuentran en z . Las funciones evaluadas fueron las que se encuentran en la Tabla del Apéndice B.

Se presenta el procedimiento que se utilizó para la comparación matricial de datos numéricos. Éste es aplicado a datos generados artificialmente a partir de funciones matemáticas, pero también puede ser aplicada a cualquier fenómeno que se pueda representar en matrices, como por ejemplo grafos, ontologías, entre otros.

Creamos un *corpus* de matrices de datos numéricos provenientes de funciones matemáticas, se toma una función y se varía un valor t de manera incremental, así se produce una matriz por cada valor de t que es variado. Se repite el proceso para las funciones que se desean comparar. Posteriormente se realizan experimentos de Búsqueda, donde se escoge una matriz a buscar (*query*) y se proporciona un

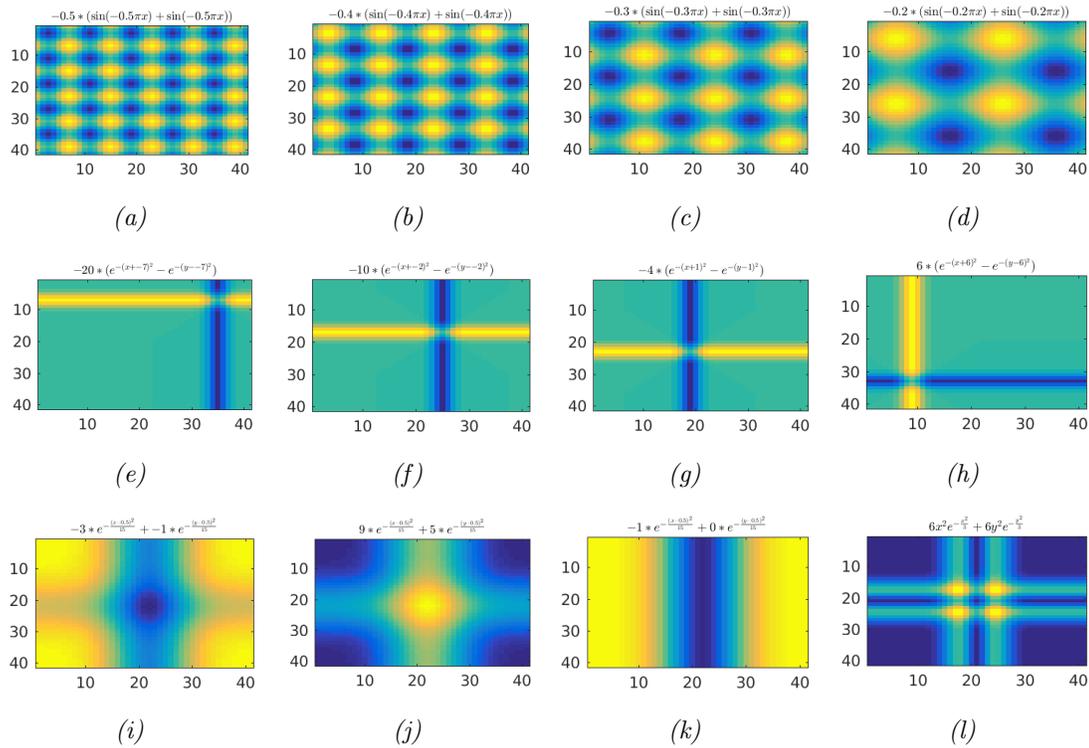


Figura 6.5: Algunas matrices del corpus.

directorio con varias matrices. En el experimento, nuestro algoritmo obtiene las matrices con distancia mínima con respecto a la matriz *query* que se escogió. Una muestra del conjunto de matrices de muestra en la Fig. 6.5, El toolbox que se utilizó para la generación del corpus se encuentra en el Apendice D. La muestra contiene tres categorías y cada categoría tiene cuatro matrices.

Las matrices fueron creadas por cada función; entonces es posible agruparlas como si fueran categorías. Se utilizó la técnica de distancia Fréchet Moderno para realizar la experimentación: primero se realizaron comparaciones entre elementos de las mismas categorías, posteriormente se realizaron comparaciones haciendo una mezcla de diferentes categorías. De estas matrices se selecciona una que será el *query*:

1. Se realiza la comparación dentro de la misma categoría que la del *query*, las

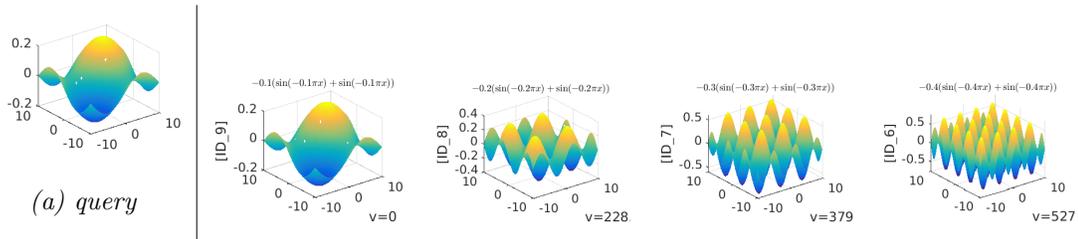


Figura 6.6: Resultados del experimento de recuperación en una sola función

funciones evaluadas para la categoría están en la Tabla 6.6, se observa el valor de la constante es variado un decimal. Los resultados se muestran en la Fig. 6.6, vemos la función que corresponde al *query*: $[-0,2(\sin(-0,2\pi x) + \sin(-0,2\pi x))]$, las mas cercanas son las que tienen el valor pequeño en las constantes de las funciones evaluadas.

Tabla 6.6: Matrices de la misma función

ID	Cercanas
[ID_9]	$-0,2 * (\sin(-0,2\pi x) + \sin(-0,2\pi x))$
[ID_8]	$-0,3 * (\sin(-0,3\pi x) + \sin(-0,3\pi x))$
[ID_7]	$-0,4 * (\sin(-0,4\pi x) + \sin(-0,4\pi x))$
[ID_6]	$-0,5 * (\sin(-0,5\pi x) + \sin(-0,5\pi x))$

- Se realiza una comparación considerando matrices de otras categorías. La Fig. 6.7 muestra en la primera columna la matriz *query* seleccionada, mientras que las otras tres columnas son las que corresponden a las matrices más cercanas. Se muestra en la parte superior la función que se evaluó para obtener esa matriz. Se puede observar que las matrices cercanas corresponden a las matrices de la misma categoría del *query*.

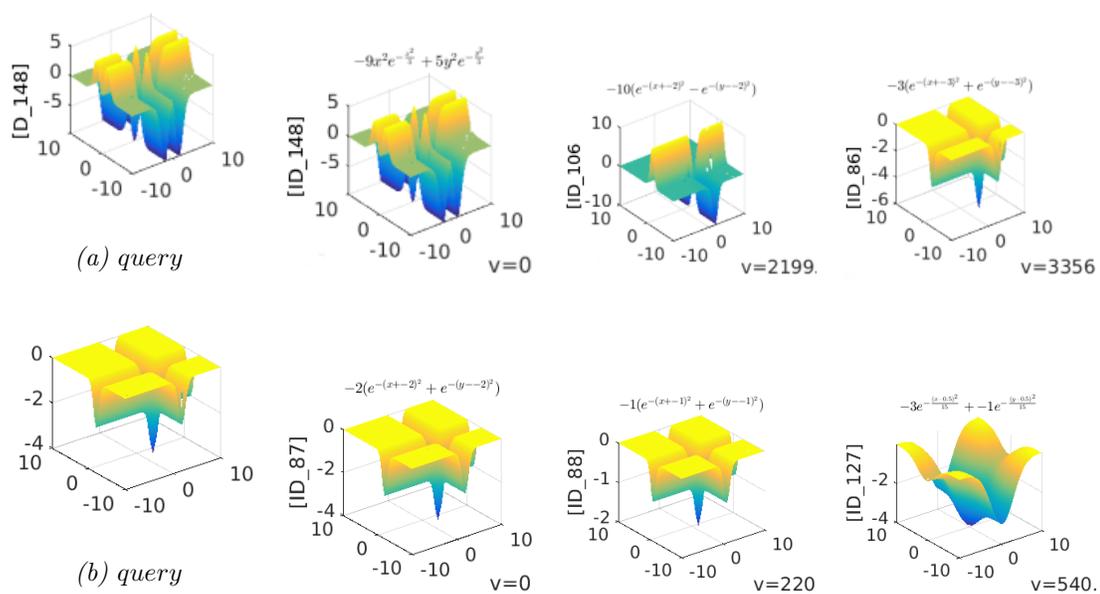


Figura 6.7: Resultado de la búsqueda con elementos mezclados.

Capítulo 7

Conclusión

En base a los objetivos de este trabajo, se presentan las conclusiones del mismo, se exponen también las contribuciones, publicaciones y trabajo futuro.

7.1. Conclusiones

Conocimiento es un tema que abre debate en importantes áreas de la ciencia, se busca obtenerlo de la manera íntegra, se intenta representarlo de alguna manera para que las computadoras sean capaces de entender una porción de nuestra realidad, de nuestro mundo. Al ser humano le conviene aproximar el conocimiento, para que una computadora sea capaz de realizar tareas cotidianas o tareas críticas que al ser humano se le dificulten. Por estas razones y otras, se intenta entender cómo es que los humanos almacenan y recuperar el conocimiento adquirido.

El significado o conocimiento puede ser almacenado como una Red Semántica Natural n -dimensional, donde los nodos son conceptos, imágenes, olores, etc. Sirven como fuente directa de conocimiento en un dominio, la red en sí, puede realizar operaciones.

Las redes semánticas naturales son una herramienta poderosa para obtener el

significado de un mundo acotado. Son poseedoras de una fuerte teoría basada en la cognición humana, es por eso que utilizarlas como medio para extraer conocimiento humano es un método factible y bien sustentado.

La metodología de elicitación presentada es una manera de obtener conocimiento humano en un dominio acotado, ésta metodología en conjunto con la distancia elástica Fréchet Moderno brindan una manera de cuantificar conocimiento humano.

Es una técnica de distancia elástica y es invariante a deformaciones, el experimento con los datos artificiales I, se muestra una deformación controlada, los resultados indican que las deformaciones mas alejadas son las que tienen una mayor distancia con respecto al query.

La técnica no considera un análisis de características, como hacen otras técnicas sino que hace una comparación global con los datos completos de las matrices. Los resultados del experimento con los datos artificiales II muestran que en un ambiente controlado las matrices recuperadas corresponden a las de la misma función.

7.2. Contribuciones

- Una técnica para representar conocimiento humano. La representación matricial de ontologías es una manera de almacenar y utilizar el conocimiento varios grupos de personas.
- Una técnica de medida elástica para matrices de datos, se desarrolló una técnica de distancia entre objetos matriciales con valores numéricos, llamada Fréchet Moderno. Ésta realiza un recorrido sobre las matrices de entrada haciendo una comparación elástica. La técnica responde a las propiedades de la métrica.
- Uso de la técnica de distancia propuesta para la comparación de conocimiento humano. Se obtuvo la representación matricial de conocimiento de varios grupos de personas con la idea de poder compararlos. La comparación fue exitosa, dos

grupos se compararon contra el grupo experto, observando la distancia entre éstos.

- Estudio del modelo de conocimiento de varios grupos reales, al obtener el conocimiento de un grupo acotado a un dominio en específico, se hace un análisis de los conceptos definidores más importantes y que dan pauta a verificar cual de todos éstos se encuentran en otros grupos. Marcando así la diferencia entre grupos.
- Análisis de la técnica de distancia usando datos artificiales e imágenes. Se crearon dos conjuntos de datos, uno es formado por funciones matemáticas, mientras que el otro se refiere a los protocolos utilizados en el proceso de obtención de las redes semánticas naturales.

7.3. Publicaciones

[Mandujano-García *et al.*, 2016] Mandujano-García, A., Figueroa, J., y Calvo, H. (2016). Comparación Cuantitativa de Matrices de Datos. En CORE 2016, CIC IPN. Ciudad de México, México.

7.4. Trabajo futuro

El trabajo realizado a esta investigación abre pauta a la continuación de nuevas investigaciones. Primero es posible empezar realizando validaciones meticulosas de la técnica de distancia, se pueden utilizar otros contextos (texto, documentos, entidades matriciales, etc).

Tratar de reducir el tiempo de ejecución del algoritmo, la complejidad computacional del algoritmo propuesto en la técnica de distancia Fréchet Moderno tiene una complejidad alta, pero mediante algunas optimizaciones al método de programación

dinámica es posible reducir la complejidad, también es posible realizar ajustes para que se utilice programación en paralelo.

Por la perspectiva del modelo de conocimiento, es posible identificar deficiencias conceptuales específicas en grupos, las deficiencias pueden estar en el sentido de las definidoras que un grupo experto tenga sobre las faltantes en los grupos no expertos. Una vez identificadas estas deficiencias, es posible proponer un ajuste en la red de conocimiento de los grupos, de tal manera se parezcan más al grupo experto.

Dicho ajuste, más adelante puede ser tomado en cuenta para un mejor aprendizaje del dominio en nuevas generaciones de individuos.

Apéndice A

Cuestionario de aplicación

Edad: _____ Genero: M F Escolaridad: _____

INSTRUCCIONES:

Lee atentamente cada una de las actividades y responde a cada cuestión. Atención, no existen respuestas correctas o incorrectas, siéntete en la libertad de responder lo que tu creas más adecuado.

Actividad I. Escribe de 5 a 10 palabras QUE DEFINAN a cada uno de los siguientes conceptos.

Ejemplo:

Manzana	() Roja	() Dulce	() Fruta	() . . .
----------------	----------	-----------	-----------	-----------

Estructura	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Memoria	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Red	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Sistema Operativo	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Información	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Proceso	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Complejidad	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Algoritmo	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Lenguaje	() _____	() _____	() _____	() _____
	() _____	() _____	() _____	() _____
	() _____			

Apuntador	()	()	()	()
	()	()	()	()
	()	()	()	()
Optimizar	()	()	()	()
	()	()	()	()
	()	()	()	()
GPU	()	()	()	()
	()	()	()	()
	()	()	()	()
CPU	()	()	()	()
	()	()	()	()
	()	()	()	()
ALU	()	()	()	()
	()	()	()	()
	()	()	()	()
USB	()	()	()	()
	()	()	()	()
	()	()	()	()
Kernel	()	()	()	()
	()	()	()	()
	()	()	()	()
URL	()	()	()	()
	()	()	()	()
	()	()	()	()
BIOS	()	()	()	()
	()	()	()	()
	()	()	()	()

Actividad II. Jerarquiza las palabras que utilizaste en cada concepto de mayor a menor. Colocando "1" a la palabra que consideras que define mejor al concepto y así sucesivamente.

Ejemplo:

Manzana	(2) Roja	(3) Dulce	(1) Fruta	(4) . . .
----------------	------------	-------------	-------------	-------------

Apéndice B

Formulas Matemáticas

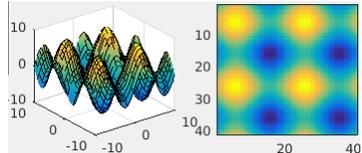
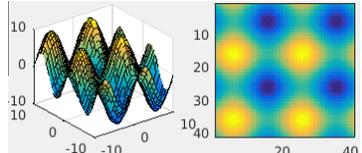
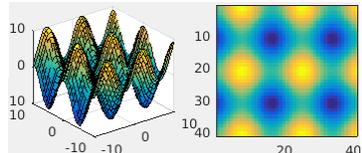
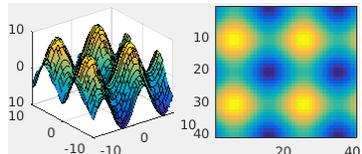
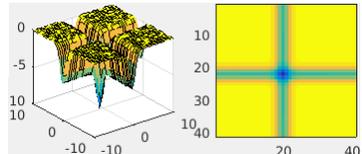
Función	Rangos	Gráfica de la Función
$a * (\sin(b\pi x) + \sin(b\pi x))$	$a, b = [-0,9, 0,9]$	
$a * (\sin(b\pi x) - \sin(b\pi x))$	$a, b = [-0,9, 0,9]$	
$a * (\sin(b\pi x) + \cos(b\pi x))$	$a, b = [-0,9, 0,9]$	
$a * (\sin(b\pi x) - \cos(b\pi x))$	$a, b = [-0,9, 0,9]$	
$a * (e^{-(x+b)^2} + e^{-(y-b)^2})$	$a = [-15, 15], b = [-7, 7]$	

Tabla B.1: Información relevante para la generación de matrices de datos artificiales (I)

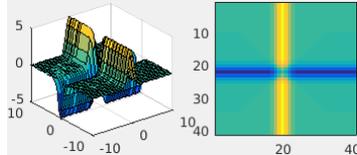
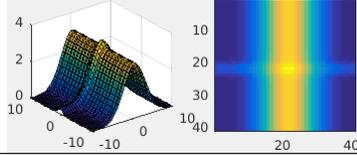
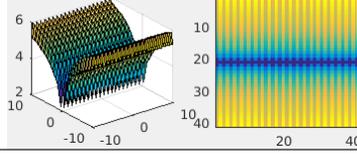
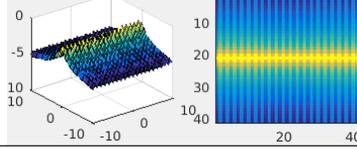
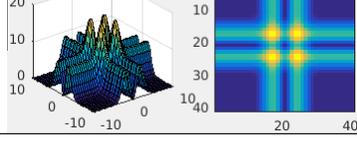
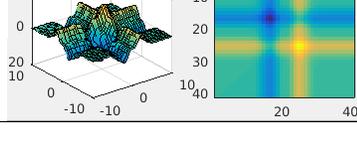
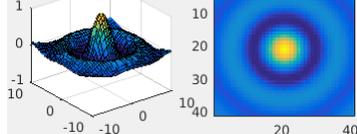
Función	Rango de Parámetros	Gráfica de la Función
$a(e^{-(x+b)^2} - e^{-(y-b)^2})$	$a = [0, inf], b = [-7, 7]$	
$ae^{-\frac{(x-0,5)^2}{15}} + b * e^{-\frac{(y-0,5)^2}{15}}$	$a = [-15, 15], b = [-7, 7], c = [0, 30]$	
$c \sin(a\pi x^2) + \log(b\pi y)$	$a = [-10, 10], b = [1, 15], c = [-1, 1]$	
$c \sin(a\pi x^2) - \log(b\pi y)$	$a = [-10, 10], b = [1, 15], c = [-1, 1]$	
$ax^2e^{-\frac{x^2}{3}} + by^2e^{-\frac{y^2}{3}}$	$a, b = [-30, 30]$	
$ax^3e^{-\frac{x^2}{3}} + by^3e^{-\frac{y^2}{3}}$	$a, b = [-30, 30]$	
$50(\frac{\sin(a\sqrt{x^2+y^2+c})}{b\sqrt{x^2+y^2}})$	$a = [-15, 15], b = [-10, 10], c = [-10, 10]$	

Tabla B.2: Información relevante para la generación de matrices de datos artificiales (II)

Apéndice C

Herramienta de comparación matricial

Esta es herramienta de apoyo para la comparación matricial. Es posible comparar dos matrices de datos de algún dominio en específico y someter esa comparación utilizando diferentes técnicas de distancia.

C.1. Herramienta de comparación matricial

Este es el manual para la ejecución de la Herramienta de Comparación Matricial. Esta herramienta es un *Toolbox* de Matlab que facilita la comparación matricial con diferentes técnicas:

1. Distancia Euclidiana
2. Bias
3. Coeficiente de Correlación
4. DIV
5. ERGAS
6. q

7. RASE
8. RMSE
9. Información Mutua
10. Fréchet Moderno

La vista principal de la herramienta es la mostrada en la Fig. C.1.

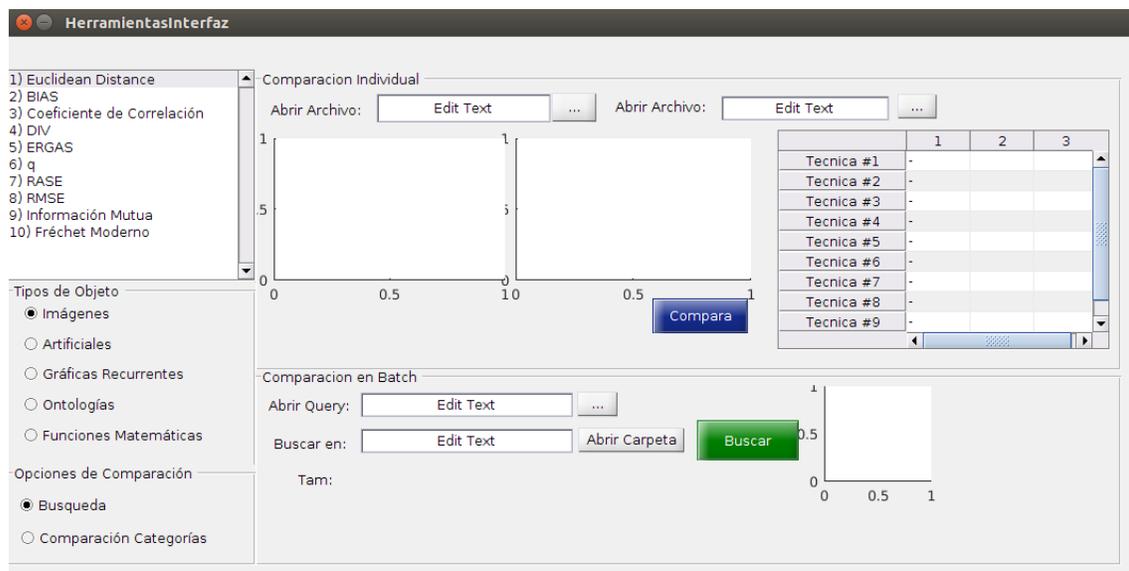


Figura C.1: Diagrama General de la herramienta

C.2. Organización de las carpetas

El Toolbox desarrollado se compone de las siguientes carpetas:

- **CÓDIGO FUENTE:** carpeta que guarda los archivos fuente, en esta ubicación es donde se encuentran los archivos necesarios, escritos en Matlab para el funcionamiento del Toolbox.

- **CONJUNTO DE DATOS:** Dentro de esta carpeta se encuentran todos los archivos de datos que han servido para hacer las pruebas de la técnica, se encuentran IMÁGENES, DATOS ARTIFICIALES, ONTOLOGÍAS.
- **DOCUMENTACIÓN:** se encuentran los archivos necesarios para compilar el presente reporte, escrito en latex, utilizando TexLive en Ubuntu Linux.
- **EXPERIMENTOS:** guarda en diferentes fechas, los archivos y resultados de los experimentos realizados.

La carpeta más importante es CÓDIGO FUENTE, dentro de ésta se encuentra el archivo principal HerramientasInterfaz.m, archivo que debe ser abierto en Matlab. Dentro de esta misma carpeta, se encuentran otras carpetas de utilidad para el análisis de diferentes técnicas.

En la Figura C.2, se muestra que el archivo HerramientasInterfaz.m tiene asociado un archivo .fig, este contiene todos los componentes visuales (botones, etiquetas, etc.) de la interfaz gráfica y solo debe ser modificado para agregar o quitar nuevos componentes. También se observa que el archivo principal hace uso de otros módulos “Misc” y “Técnicas”, en el primero se tienen scripts para el funcionamiento de la interfaz o preprocesamiento de las matrices. En el segundo se tienen los scripts para calcular diferentes técnicas de comparación matricial, que son incorporados a la herramienta para ver diferencias entre éstas. Dentro del segundo modulo, se encuentran los archivos desarrollados para la técnica de distancia matricial llamada **Fréchet Moderno**.

C.3. Comparación de objetos matriciales

La herramienta considera evaluar diferentes técnicas de distancia matricial para diferentes objetos de dominios acotados, es decir, se pueden comparar elementos de conjuntos de datos de:

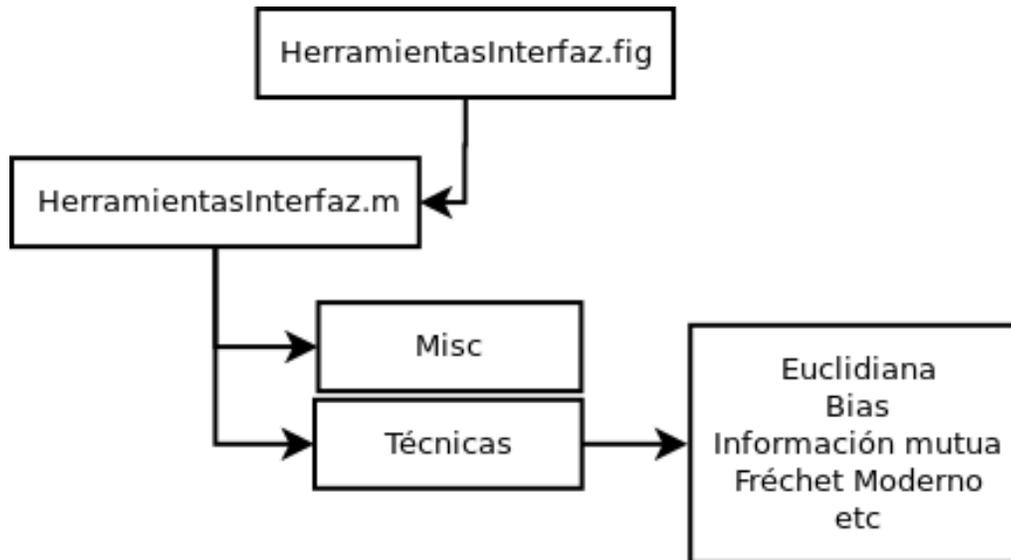


Figura C.2: Diagrama General de la herramienta

- Imágenes
- Datos artificiales I
- Matrices de análisis recurrentes
- Ontologías
- Funciones Matemáticas

Para poder comparar objetos matriciales primero debemos escoger la técnica que deseamos evaluar, posteriormente se tendrá que escoger el tipo de objetos matriciales (imágenes, datos artificiales, ontologías, etc.).

Una vez realizado estos dos pasos, se pueden realizar dos tareas:

1. Comparar dos objetos matriciales (Comparación Individual)
2. Comparar un objeto (*query*) contra un directorio que contiene varios objetos del mismo tipo que el *query* (Comparación en batch)

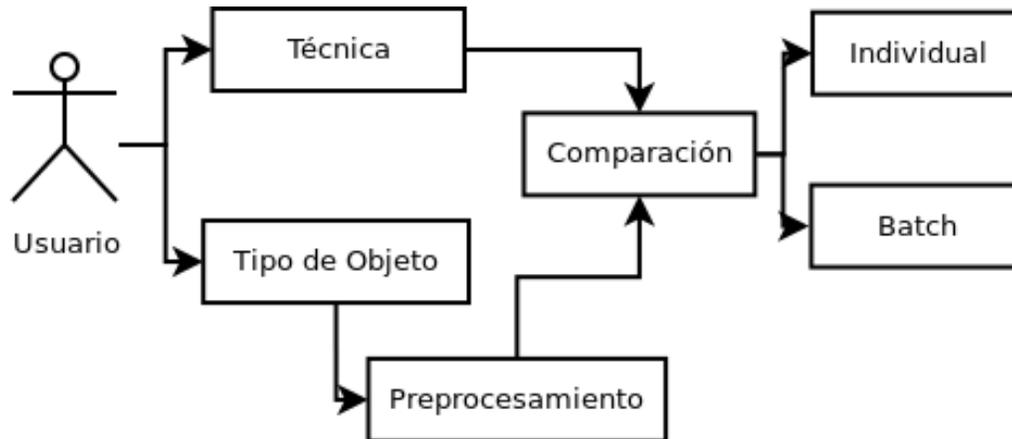


Figura C.3: Diagrama de flujo de la comparación

C.3.1. Ejecución de la herramienta

Para empezar la ejecución de la herramienta se necesita cargar el archivo HerramientasInterfaz.m en el editor de Matlab y posteriormente, ejecutar con la tecla F5. Se visualizará la interfaz de la Fig. C.1.

En la Fig. C.4 se tiene la interfaz visual de la comparación individual, en ese caso, se escogió como objetos: imágenes, se ejecutó con varias técnicas, euclidiana, RMSE, q.

Se escogen los objetos a comparar, en este caso deben ser imágenes las dos, posteriormente se presiona el boton azul Comparar.

En la Fig. C.5 se muestra la configuración para realizar una comparación en batch. Lo primero que se realiza es escoger la técnica y el tipo de objetos que se utilizaran. Posteriormente se presiona el botón para abrir el *query*, se busca y se visualizará en la parte derecha de la sección. Después se debe cargar el directorio donde se encuentren los objetos. Una vez cargado el directorio, se mostrará información de los objetos que contiene, y ya se podrá realizar la tarea de recuperación presionando en el botón verde, buscar.

La herramienta organiza el resultado de la búsqueda y lo muestra en una serie de

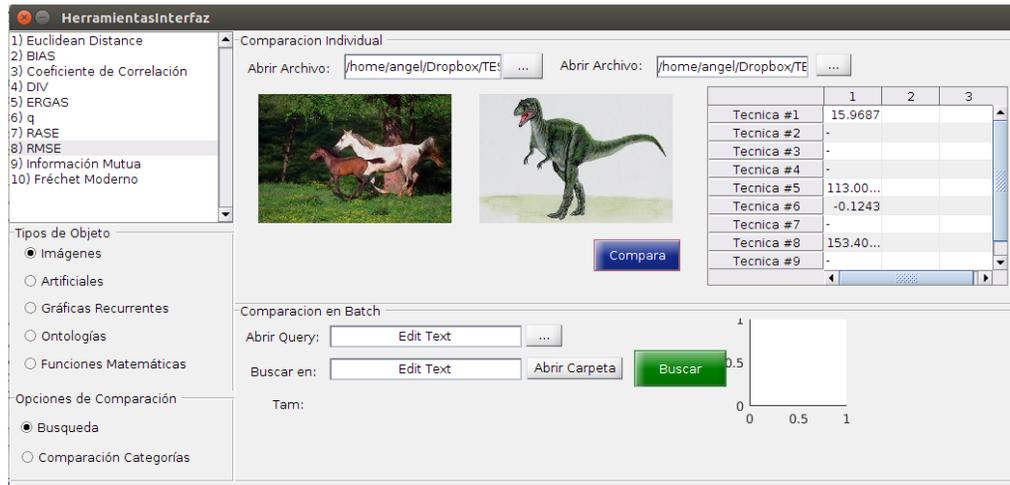


Figura C.4: Comparación Individual

gráficas, éstas gráficas se encuentran de menor a mayor de tal manera que se vean cuales fueron los objetos más cercanos al *query*. Como se muestra en la Fig. C.6

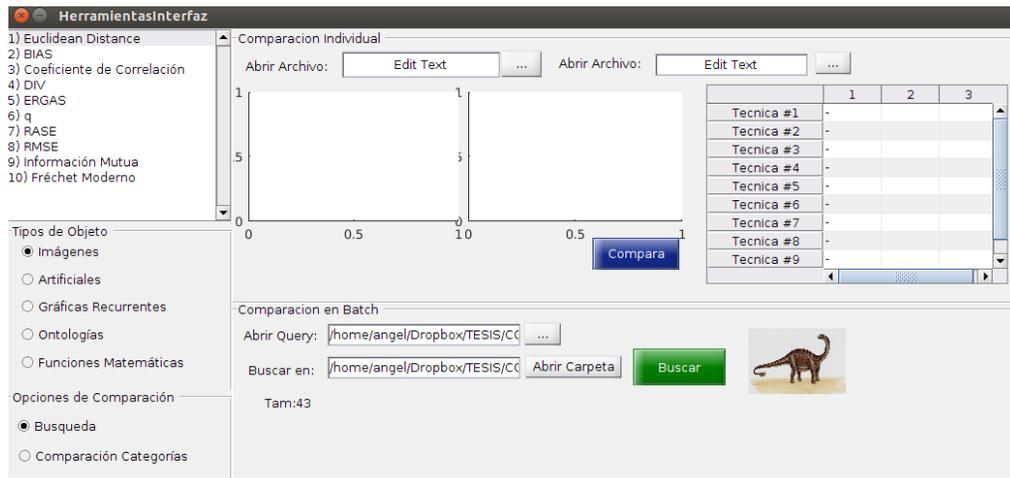


Figura C.5: Comparación en batch

C.4. Detalles Técnicos

Para algunos objetos es necesario realizar un preprocesamiento, en particular a las imágenes:

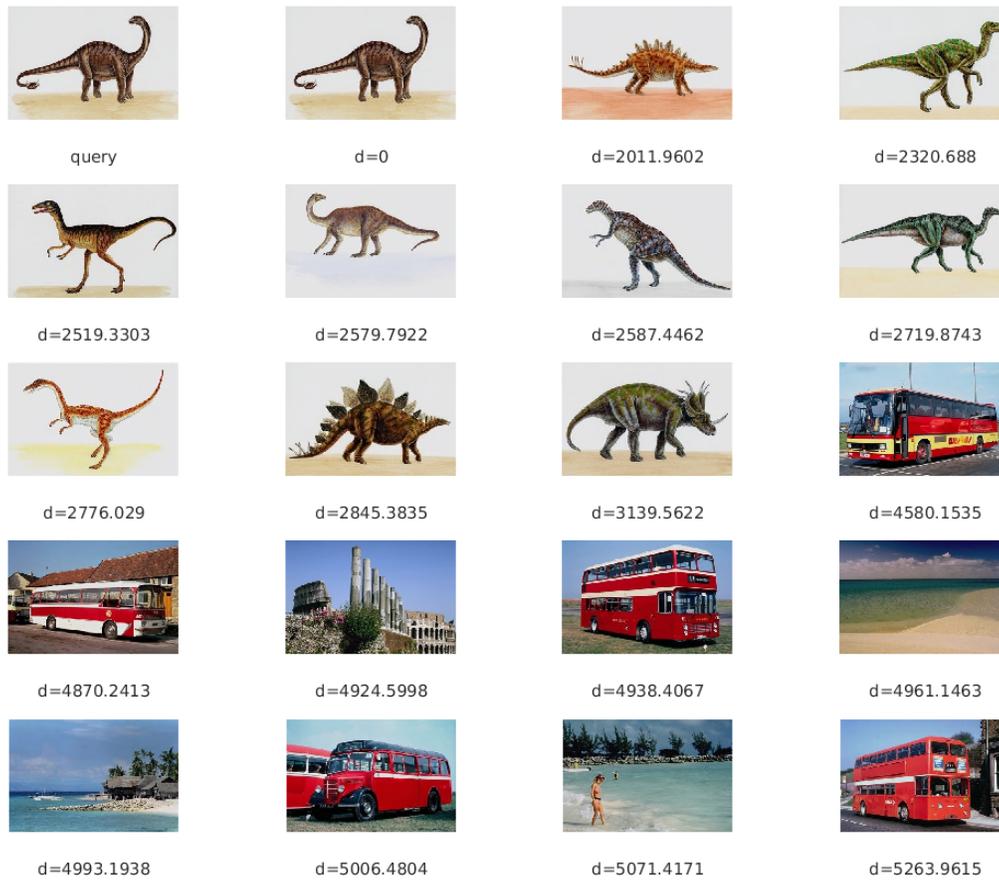


Figura C.6: Resultados de la comparación en batch usando imágenes

- Se redimensiona la imagen haciendo un muestreo. En el archivo `misc/inicializaImage.m` se puede configurar el muestreo, solo es necesario descomentar una línea para que cambie el tamaño de la imagen.
- De los canales a 8 bits de la imagen (RGB), solo se toma uno, teniendo una sola matriz con valores entre $[0 - 255]$.

C.5. Sobre Matlab

La herramienta desarrollada para la comparación matricial ha sido desarrollada en Matlab versión R2016a 64 bits, bajo un ambiente linux, ubuntu 10-2015, es posible

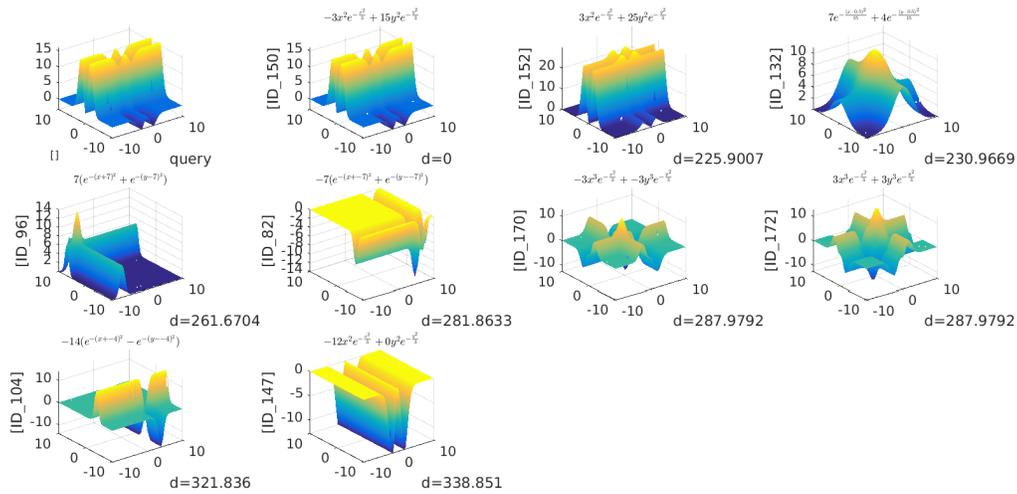


Figura C.7: Resultados de la comparación en batch usando funciones matemáticas

que algunas características dentro de Matlab no sean consistentes en otras versiones de Matlab o de sistemas operativos. Los detalles técnicos de la computadora principal de desarrollo se muestran a continuación:

- Disco duro: 720Gb
- Memoria RAM: 4Gb
- S.O.: Ubuntu 64bits

Se ha seleccionado Matlab como herramienta de desarrollo debido a que ofrece características importantes, la implementación de los componentes desarrollados como una colección reutilizable, conjunto de interfaces gráficas. Estas características son importantes en el ambiente académico-científico porque el código fuente es interpretado y brinda la facilidad de distribuir y reutilizar el código, también permite la adaptación de los procedimientos y las interfaces desarrolladas.

C.6. Código fuente

Listing C.1: HerramientasInterfaz.m

```

1 function varargout = HerramientasInterfaz(varargin)
2 %Begin initialization code – DO NOT EDIT
3 gui_Singleton = 1;
4 gui_State = struct('gui_Name',       mfilename, ...
5                   'gui_Singleton',  gui_Singleton, ...
6                   'gui_OpeningFcn', @HerramientasInterfaz_OpeningFcn, ...
7                   'gui_OutputFcn',  @HerramientasInterfaz_OutputFcn, ...
8                   'gui_LayoutFcn',  [], ...
9                   'gui_Callback',    []);
10 if nargin && ischar(varargin{1})
11     gui_State.gui_Callback = str2func(varargin{1});
12 end
13
14 if nargin
15     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
16 else
17     gui_mainfcn(gui_State, varargin{:});
18 end
19 %End initialization code – DO NOT EDIT
20
21 function HerramientasInterfaz_OpeningFcn(hObject, eventdata, ...
22     handles, varargin)
23     handles.output = hObject;
24     guidata(hObject, handles);
25
26     pathc = mfilename('fullpath'); %%obtiene el directorio y ...
27     script actual
28     [currentpath, filenames, fileextension]= fileparts(pathc);
29
30     %%AGREGA SUBDIRECTORIOS
31     addpath(strcat(currentpath, '/Tecnicas/Euclidiana'));
32     addpath(strcat(currentpath, '/Tecnicas/Frechet'));
33     addpath(strcat(currentpath, '/MISC'));
34     addpath(strcat(currentpath, '/Tecnicas/Image_Index_Analys'));
35     addpath(strcat(currentpath, '/Tecnicas/Mutual_Information'));
36
37     %% Configuracion inicial para la tabla de las tecnicas
38     tableData = cell(10, 3);
39     for n = 1 : size(tableData, 1)
40         rowHeaders{n} = sprintf('Tecnica #%d', n);
41         tableData{n,1} = '-';
42     end
43     set(handles.uitable1, 'RowName', rowHeaders);
44     set(handles.uitable1, 'ColumnWidth', {60, 60, 60});
45     set(handles.uitable1, 'data', tableData);

```

```

45     % Guarda elementos importantes en los handles de matlab
46     handles.currentpath = currentpath;
47     tipoObjetos = 0;
48     handles.tipoObjetos = tipoObjetos;
49     guidata(hObject, handles);
50
51     function varargout = HerramientasInterfaz_OutputFcn(hObject, ...
52         eventdata, handles)
53         varargout{1} = handles.output;
54
55     function edit3_Callback(hObject, eventdata, handles)
56
57     function edit3_CreateFcn(hObject, eventdata, handles)
58         if ispc && isequal(get(hObject, 'BackgroundColor'), ...
59             get(0, 'defaultUicontrolBackgroundColor'))
60             set(hObject, 'BackgroundColor', 'white');
61         end
62
63     function edit2_Callback(hObject, eventdata, handles)
64
65     function edit2_CreateFcn(hObject, eventdata, handles)
66         if ispc && isequal(get(hObject, 'BackgroundColor'), ...
67             get(0, 'defaultUicontrolBackgroundColor'))
68             set(hObject, 'BackgroundColor', 'white');
69         end
70
71     function listbox1_Callback(hObject, eventdata, handles)
72
73     function listbox1_CreateFcn(hObject, eventdata, handles)
74         if ispc && isequal(get(hObject, 'BackgroundColor'), ...
75             get(0, 'defaultUicontrolBackgroundColor'))
76             set(hObject, 'BackgroundColor', 'white');
77         end
78
79     %—— COMPARACION BATCH
80     %—— BOTON PARA CARGAR EL QUERY
81     function pushbutton1_Callback(hObject, eventdata, handles)
82         currentpath = handles.currentpath;
83         fp = fileparts(currentpath);
84         [x,y]=meshgrid(-10:.5:10);
85
86         if handles.tipoObjetos == 0 % solo imagenes
87             [filename, pathname] = ...
88                 uigetfile(strcat(fp, '/*.jpg'), 'Busca Imagenes');
89             dirQuery = strcat(pathname, filename);
90             query = imread(dirQuery);
91             imshow(query, 'InitialMagnification', 'fit', 'parent', ...
92                 handles.axesimage1);
93             query = inicializaImage(query);

```

```

88         set(handles.edit2, 'String', dirQuery);
89
90     elseif handles.tipoObjetos == 4 % funciones matematicas
91         [filename, pathname] = ...
92             uigetfile(strcat(fp, '/*.mat'), 'Busca matriz');
93         dirQuery = strcat(pathname, filename);
94         query = AbrirDatos_MAT(filename, pathname);
95         query = query.M;
96
97         surf(x,y,query, 'parent', handles.axesimage1);
98         axis tight
99         shading flat
100        shading interp
101        rotate3d on
102        set(handles.edit2, 'String', dirQuery);
103
104    else % otros objetos
105        [filename, pathname] = ...
106            uigetfile(strcat(fp, '/*.mat'), 'Busca matriz');
107        dirQuery = strcat(pathname, filename);
108        query = AbrirDatos_MAT(filename, pathname);
109        query = query.M;
110
111        imagesc(query, 'parent', handles.axesimage1);
112        set(handles.edit2, 'String', dirQuery);
113    end
114
115    handles.dirQuery = dirQuery;
116    handles.query = query;
117    guidata(hObject, handles);
118
119 %—— CARGA LA CARPETA DE BUSQUEDA
120 function pushbutton4_Callback(hObject, eventdata, handles)
121     currentpath = handles.currentpath;
122     fp = fileparts(currentpath);
123
124     directory = uigetdir(fp, 'Buscar');
125     allFiles = dir(directory);
126     fileList = {allFiles(~[allFiles.isdir]).name};
127     [n m] = size(fileList);
128
129     if handles.tipoObjetos == 0
130         for i = 1: m
131             ruta = strcat(directory, '/', char(fileList(i)));
132             nombres{i} = strcat('obj', int2str(i));
133             img = imread(ruta);
134             imgBrutas{i} = img;
135             img = inicializaImage(img);
136             tam = size(img,2);

```

```

135         matrices{i} = img;
136     end
137     objetosBruto = containers.Map(nombres, imgBrutas);
138     handles.objetosBruto = objetosBruto;
139
140     elseif handles.tipoObjetos == 4
141         for i = 1: m
142             nombres{i} = strcat('obj', int2str(i));
143             elemento = AbrirDatos_MAT(char(fileList(i)), directory);
144             elemento = elemento.M;
145             matrices{i} = elemento;
146             tam = size(elemento,2);
147         end
148     else
149         for i = 1: m
150             % arreglo con los nombres de los archivos
151             nombres{i} = strcat('obj', int2str(i));
152             elemento = AbrirDatos_MAT(char(fileList(i)), directory);
153             elemento = elemento.M;
154             matrices{i} = elemento; % arreglo con la matriz
155             tam = size(elemento,2);
156         end
157     end
158
159     set(handles.edit5, 'String', directory);
160     set(handles.text25, 'String', strcat('Tam:', int2str(tam)));
161
162     %map nombre como llave y la matriz como valor
163     objetos = containers.Map(nombres, matrices);
164
165     handles.fileList = fileList;
166     handles.directory = directory;
167     handles.objetos = objetos;
168     guidata(hObject, handles);
169
170     %—— BOTON DE COMPARAR Y REALIZAR LA BUSQUEDA
171     function pushbutton3_Callback(hObject, eventdata, handles)
172         tecnicaIndex = get(handles.listbox1, 'Value');
173         disp(['Tecnica Seleccionada: ', int2str(tecnicaIndex)]);
174
175         fileList = handles.fileList;
176         objetos = handles.objetos;
177         query = handles.query;
178
179         %hace la busqueda
180         tic;
181         [ceranosValues, cercanosObjs, objsTags] = ...
182             Busqueda(tecnicaIndex, query, objetos, fileList);
183         n = toc;

```

```

183     n = (n/60)/60
184
185     %MOSTRAR RESULTADO INTERFAZ
186     figure1 = figure('units','normalized','outerposition',[0 0 1 ...
187                    1]);
187     if handles.tipoObjetos == 0 %%imágenes
188         objetosBruto = handles.objetosBruto;
189
190         % query
191         subplot(5,4,1);
192         obje = cercanosObjs(1);
193         obj = cell2mat(values(objetosBruto, obje));
194         imshow(obj);
195         xlabel('query');
196
197         % cercanas
198         for i = 1: size(cercanosValues,2)
199             obje = cercanosObjs(i);
200             valor = cercanosValues(i);
201             obj = cell2mat(values(objetosBruto, obje));
202
203             subplot(5,4,i+1);
204             imshow(obj);
205             xlabel(strcat('d=',num2str(valor)))
206         end
207     elseif handles.tipoObjetos == 4
208
209         [x,y]=meshgrid(-10:.5:10);
210
211         directory = handles.directory;
212         tmp = ...
213             load('-~/Funciones_Matematicas/DATA/Matrices/names.mat');
214         names = tmp.('namesAdded');
215
216         % muestra el query
217         subplot(5,4,1);
218         surf(x,y,query)
219         axis tight
220         shading flat
221         rotate3d on
222         xlabel('query');
223         valor = cercanosValues(1);
224         ylabel(strcat('[' , valor , ']'),'fontsize',8);
225
226         for i = 1: size(cercanosValues,2)
227             subplot(5,4,i+1);
228
229             cercano = cercanosObjs(i);

```

```

230         valor = cercanosValues(i);
231         id = regexp(objsTags{i}, 'ID-[0-9]*', 'match');
232         label = names(id{1});
233         mtrx = cell2mat(values(objetos, cercano));
234
235         surf(x,y,mtrx)
236         axis tight
237         shading flat
238         shading interp
239         rotate3d on
240         title(label, 'Interpreter', 'latex', 'fontsize', 8);
241         xlabel(strcat('d=', num2str(valor)))
242         id_1 = regexp(id{1}, '-', '\\\\-');
243         zlabel(strcat('[' , id_1 , ']'), 'fontsize', 11);
244         xt = get(gca, 'XTick');
245
246     end
247 else % casos no especiales
248
249     % query
250     subplot(5,4,1);
251     imagesc(query);
252     xlabel('query');
253
254     % cercanas
255     for i = 1: size(cercanosValues,2)
256         obje = cercanosObjs(i);
257         valor = cercanosValues(i);
258         obj = cell2mat(values(objetos, obje));
259
260         subplot(5,4,i+1);
261         imagesc(obj);
262         xlabel(strcat('d=', num2str(valor)))
263     end
264 end
265
266 %CREAR IMAGEN DE MUESTRAS
267 fileplot1 = ...
268     strcat('~/Funciones_Matematicas/DATA/Experimentos', '/file.png');
269 saveas(figure1, fileplot1);
270 disp('done!');
271
272 function edit4_Callback(hObject, eventdata, handles)
273
274 function edit4_CreateFcn(hObject, eventdata, handles)
275     if ispc && isequal(get(hObject, 'BackgroundColor'), ...
276         get(0, 'defaultUicontrolBackgroundColor'))
277         set(hObject, 'BackgroundColor', 'white');
278     end

```

```

277
278 %—— COMPARACION INDIVIDUAL
279 %—— COMPARACION INDIVIDUAL, PRIMER OBJETO
280 function pushbutton9_Callback(hObject, eventdata, handles)
281     currentpath = handles.currentpath;
282     fp = fileparts(currentpath);
283
284     if handles.tipoObjetos == 0
285         [filename, pathname] = ...
286             uigetfile(strcat(fp, '/*.jpg'), 'Busca Imagenes');
287         matrizDir = strcat(pathname, filename);
288         matrizOne = imread(matrizDir);
289         imshow(matrizOne, 'InitialMagnification', 'fit', ...
290             'parent', handles.matriz1);
291         matrizOne = inicializaImage(matrizOne);
292     else
293         [filename, pathname] = ...
294             uigetfile(strcat(fp, '/*.mat'), 'Busca matriz');
295         matrizDir = strcat(pathname, filename);
296         matrizOne = AbrirDatos_MAT(filename, pathname);
297         matrizOne = matrizOne.M;
298         imagesc(matrizOne, 'parent', handles.matriz1);
299         set(handles.edit2, 'String', matrizDir);
300     end
301
302     set(handles.edit4, 'String', matrizDir);
303     handles.matrizDir = matrizDir;
304     handles.matrizOne = matrizOne;
305     guidata(hObject, handles);
306
307 %—— COMPARACION INDIVIDUAL, SEGUNDO OBJETO
308 function pushbutton2_Callback(hObject, eventdata, handles)
309     currentpath = handles.currentpath;
310     fp = fileparts(currentpath);
311
312     if handles.tipoObjetos == 0
313         [filename, pathname] = ...
314             uigetfile(strcat(fp, '/*.jpg'), 'Busca Imagenes');
315         matriz2Dir = strcat(pathname, filename);
316         matrizTwo = imread(matriz2Dir);
317         imshow(matrizTwo, 'InitialMagnification', 'fit', ...
318             'parent', handles.matriz2);
319         matrizTwo = inicializaImage(matrizTwo);
320     else
321         [filename, pathname] = ...
322             uigetfile(strcat(fp, '/*.mat'), 'Busca matriz');
323         matriz2Dir = strcat(pathname, filename);
324         matrizTwo = AbrirDatos_MAT(filename, pathname);
325         matrizTwo = matrizTwo.M;

```

```

320         imagesc(matrizTwo, 'parent', handles.matriz2);
321         set(handles.edit2, 'String', matri2Dir);
322     end
323
324     set(handles.edit3, 'String', matri2Dir);
325     handles.iDir2 = matri2Dir;
326     handles.matrizTwo = matrizTwo;
327     guidata(hObject, handles);
328
329     %—— Boton COMPARACION INDIVIDUAL
330     function pushbutton8_Callback(hObject, eventdata, handles)
331         selectedIndex = get(handles.listbox1, 'Value');
332         disp(['Tecnica Seleccionada: ', int2str(selectedIndex)]);
333
334         selectedObj = handles.tipoObjetos;
335         disp(['objetos tipo: ', int2str(selectedObj)]);
336
337         comparacion = 0;
338         matrizOne = handles.matrizOne;
339         matrizTwo = handles.matrizTwo;
340
341         % se escoge la herramienta que se va a utilizar
342         tic
343         comparacion = ComparacionIndividual(selectedIndex, ...
344             matrizOne, matrizTwo);
345         toc
346         % actualizar tabla
347         oldData = get(handles.uitable1, 'Data');
348         oldData{selectedIndex, 1} = comparacion;
349         newData = oldData;
350         set(handles.uitable1, 'Data', newData);
351         disp(['done']);
352     function uipanel13_SelectionChangeFcn(hObject, eventdata, handles)
353         % 0 = imagenes
354         % 1 = matrices a mano
355         % 2 = matrices de recurrencia
356         % 3 = ontologias
357         % 4 = funciones matematicas
358         tipoObjetos = 0;
359         if hObject == handles.rbImagenes
360             tipoObjetos = 0;
361         elseif hObject == handles.rbMatricesMano
362             tipoObjetos = 1;
363         elseif hObject == handles.rbGraficasRecurrencia
364             tipoObjetos = 2;
365         elseif hObject == handles.rbOntologias
366             tipoObjetos = 3;
367         elseif hObject == handles.rbFunciones

```

```

368         tipoObjetos = 4;
369     end
370     handles.tipoObjetos = tipoObjetos;
371     guidata(hObject, handles);
372
373 function uipanel14_SelectionChangeFcn(hObject, eventdata, handles)
374     tipoBatch = 0;
375     if hObject == handles.rbBusqueda
376         tipoBatch = 0;
377         set(handles.pushbutton5, 'Visible', 'off');
378         set(handles.pushbutton6, 'Visible', 'off');
379         set(handles.pushbutton7, 'Visible', 'off');
380     elseif hObject == handles.rbCategoria
381         tipoBatch = 1;
382         set(handles.pushbutton5, 'Visible', 'on');
383         set(handles.pushbutton6, 'Visible', 'on');
384         set(handles.pushbutton7, 'Visible', 'on');
385     end
386     disp(['Batch tipo: ', int2str(tipoBatch)]);
387
388 function edit5_Callback(hObject, eventdata, handles)
389
390 function edit5_CreateFcn(hObject, eventdata, handles)
391 if ispc && isequal(get(hObject, 'BackgroundColor'), ...
392     get(0, 'defaultUicontrolBackgroundColor'))
393     set(hObject, 'BackgroundColor', 'white');
394 end

```

Listing C.2: AbrirDatosMAT.m

```

1 function s = AbrirDatos_MAT(filename, dirDatos)
2     fullMatFileName = fullfile(dirDatos, filename);
3     if ~exist(fullMatFileName, 'file')
4         message = sprintf('%s does not exist', fullMatFileName);
5         uiwait(warndlg(message));
6     else
7         s = load(fullMatFileName);
8     end
9 end

```

Listing C.3: Busqueda.m

```

1 function [Vsorted, objs, objsTags] = Busqueda(selectedIndex, ...
2     query, objetos, fileList)

```

```

3     valores = [];
4     switch selectedIndex
5         case 1
6             disp('one');    %%EUCLIDIANA
7             for i = 1: c
8                 disp(['M',int2str(i)]);
9                 idObj = strcat('obj', int2str(i));
10                nombres{i} = idObj;
11                matriz = objetos(idObj);
12                value = Euclidean2D(double(query), double(matriz));
13                %valores{i} = value;
14                valores = [valores value];
15            end
16        case 2
17            disp('two')    %%BIAS
18            for i = 1: c
19                disp(['M',int2str(i)]);
20                idObj = strcat('obj', int2str(i));
21                nombres{i} = idObj;
22                matriz = objetos(idObj);
23                value = bias_f(double(query), double(matriz));
24                %valores{i} = value;
25                valores = [valores value];
26            end
27        case 3
28            disp('three')    %%COEFICIENTE DE CORRELACION
29            for i = 1: c
30                disp(['M',int2str(i)]);
31                idObj = strcat('obj', int2str(i));
32                nombres{i} = idObj;
33                matriz = objetos(idObj);
34                value = ccc_f(double(query), double(matriz));
35                %corr2
36                %valores{i} = value;
37                valores = [valores value];
38            end
39        case 4
40            disp('four')    %%DIV
41            for i = 1: c
42                disp(['M',int2str(i)]);
43                idObj = strcat('obj', int2str(i));
44                nombres{i} = idObj;
45                matriz = objetos(idObj);
46                value = div_f(double(query), double(matriz));
47                %valores{i} = value;
48                valores = [valores value];
49            end
50        case 5
51            disp('five')    %%ERGAS

```

```

52         for i = 1: c
53             disp(['M',int2str(i)]);
54             idObj = strcat('obj', int2str(i));
55             nombres{i} = idObj;
56             matriz = objetos(idObj);
57             value = ergas_f(double(query), double(matriz),0.5);
58             %valores{i} = value;
59             valores = [valores value];
60         end
61     case 6
62         disp('six')    %%q
63         for i = 1: c
64             disp(['M',int2str(i)]);
65             idObj = strcat('obj', int2str(i));
66             nombres{i} = idObj;
67             matriz = objetos(idObj);
68             value = q_f(double(query), double(matriz));
69             %valores{i} = value;
70             valores = [valores value];
71         end
72     case 7
73         disp('seven')    %%rase
74         for i = 1: c
75             disp(['M',int2str(i)]);
76             idObj = strcat('obj', int2str(i));
77             nombres{i} = idObj;
78             matriz = objetos(idObj);
79             value = rase_f(double(query), double(matriz));
80             %valores{i} = value;
81             valores = [valores value];
82         end
83     case 8
84         disp('eight')    %%rmse
85         for i = 1: c
86             disp(['M',int2str(i)]);
87             idObj = strcat('obj', int2str(i));
88             nombres{i} = idObj;
89             matriz = objetos(idObj);
90             value = rmse_f(double(query), double(matriz));
91             %valores{i} = value;
92             valores = [valores value];
93         end
94     case 9
95         disp('nine')    %%MUTUAL INFORMATION
96         for i = 1: c
97             disp(['M',int2str(i)]);
98             idObj = strcat('obj', int2str(i));
99             nombres{i} = idObj;
100            matriz = objetos(idObj);

```

```

101         value = MutualInformation(double(query), ...
102             double(matriz));
103         %mi(a,b);
104         %valores{i} = value;
105         valores = [valores value];
106     end
107     case 10
108         disp('diez')
109         for i = 1: c
110             disp(['M',int2str(i)]);
111             idObj = strcat('obj', int2str(i));
112             nombres{i} = idObj;
113             matriz = objetos(idObj);
114             value = DIW2D(double(query), double(matriz));
115             %valores{i} = value;
116             valores = [valores value];
117         end
118     otherwise
119         disp('other value')
120     end
121 end
122 [Vsorted, SortIndex] = sort(valores);
123 objs = nombres(SortIndex);
124 objsTags = fileList(SortIndex);
125 %comparaciones = containers.Map(valores, nombres);
126 end

```

Listing C.4: ComparacionIndividual.m

```

1 function res = ComparacionIndividual(selectedIndex, A, B)
2     comparacion = -1;
3     switch selectedIndex
4         case 1
5             comparacion = Euclidean2D(A, B);
6         case 2
7             comparacion = bias_f(A, B);
8         case 3
9             comparacion = ccc_f(A, B);
10        case 4
11            comparacion = div_f(A, B);
12        case 5
13            comparacion = ergas_f(A, B,0.5);
14        case 6
15            comparacion = q_f(A, B);
16        case 7
17            comparacion = rase_f(A, B);

```

```
18     case 8
19         comparacion = rmse_f(A, B);
20     case 9
21         comparacion = MutualInformation(A, B);
22     case 10
23         comparacion = DIW2D(double(A), double(B));
24     otherwise
25         disp('other value')
26     end
27     res = comparacion;
28 end
```

Listing C.5: inicializaImage.m

```
1 function res = inicializaImage(image)
2     % Recorta la imagen a 256x256 pixeles
3     image = imresize(image, [256 256]);
4     % Obtiene un solo canal.
5     image = image(:,:,1);
6     % hace el muestreo para hacer pequena la matriz
7     % image = image(1 : 5 : end, 1 : 5 : end);      % 52x52
8     image = image(1 : 6 : end, 1 : 6 : end);      % 43x43
9     % image = image(1 : 7 : end, 1 : 7 : end);      % 37x37
10    % image = image(1 : 10 : end, 1 : 10 : end);     % 26x26
11    % image = image(1 : 15 : end, 1 : 15 : end);    % 18x18
12    % image = image(1 : 20 : end, 1 : 20 : end);    % 13x13
13    % image = image(1 : 30 : end, 1 : 30 : end);    % 9x9
14    % image = image(1 : 35 : end, 1 : 35 : end);    % 8x8
15    res = image;
16 end
```

Apéndice D

Matrix Maker

Matrix Maker es una herramienta para la creación de un cuerpo de datos sintético, el cual consiste en la evaluación de funciones matemáticas. Ha sido desarrollado para brindar soporte a investigaciones sobre medidas de similitud en matrices de datos.

El *dataset* consiste en generar matrices de datos usando funciones matemáticas, mediante la variación de algún parámetro, se consiguen así matrices diferentes. Son 200 elementos en el conjunto de datos, es decir, 200 matrices numéricas. Se evaluaron 12 funciones: las cuales contienen parámetros a , b o c que fueron variados dentro de un rango, para poder generar cada matriz. Los parámetros son variados de manera controlada, se van aumentando o disminuyendo de manera secuencial, así se sabrá de antemano, cuales son las matrices cercanas o alejadas. Conforme más grande sea el valor de un parámetro, mas lejos se encontrará.

Existe una versión, en la cual se pueden generar matrices de manera aleatoria, con el objetivo de generar matrices donde no se conozca su tendencia controlada.

D.0.1. Funciones evaluadas y rangos de evaluación

En el cuadro D.1 se presentan las matrices que han sido evaluadas con sus respectivos rangos.

Función	Rango de parámetros
$a * (\sin(b\pi x) + \sin(b\pi y))$	$a, b = [-0,9, 0,9]$
$a * (\sin(b\pi x) - \sin(b\pi y))$	$a, b = [-0,9, 0,9]$
$a * (\sin(b\pi x) + \cos(b\pi y))$	$a, b = [-0,9, 0,9]$
$a * (\sin(b\pi x) - \cos(b\pi y))$	$a, b = [-0,9, 0,9]$
$a * (e^{-(x+b)^2} + e^{-(y-b)^2})$	$a = [-15, 15], b = [-7, 7]$
$a(e^{-(x+b)^2} - e^{-(y-b)^2})$	$a = [0, inf], b = [-7, 7]$
$ae^{-\frac{(x-0,5)^2}{15}} + b * e^{-\frac{(y-0,5)^2}{15}}$	$a = [-15, 15], b = [-7, 7], c = [0, 30]$
$c \sin(a\pi x^2) + \log(b\pi y)$	$a = [-10, 10], b = [1, 15], c = [-1, 1]$
$c \sin(a\pi x^2) - \log(b\pi y)$	$a = [-10, 10], b = [1, 15], c = [-1, 1]$
$ax^2e^{-\frac{x^2}{3}} + by^2e^{-\frac{y^2}{3}}$	$a, b = [-30, 30]$
$ax^3e^{-\frac{x^2}{3}} + by^3e^{-\frac{y^2}{3}}$	$a, b = [-30, 30]$
$50\left(\frac{\sin(a\sqrt{x^2+y^2+c})}{b\sqrt{x^2+y^2}}\right)$	$a = [-15, 15], b = [-10, 10], c = [-10, 10]$

Tabla D.1: Funciones matemáticas evaluadas en los rangos especificados para cada uno de sus parámetros.

D.0.2. Funcionamiento de la herramienta

Matrix Maker es un script de Matlab, mediante una interfaz gráfica se pueden hacer configuraciones. El script se llama `matrix_maker.m`, contiene las funciones necesarias para ejecutar la interfaz gráfica de la herramienta como se muestra en la Fig. D.1.

Dentro de la interfaz es fácil escoger la función deseada, el número de evaluaciones que se desea para esa función y el directorio donde serán guardadas las matrices generadas. La herramienta muestra una vista de ejemplo de la función matemática escogida.

El directorio de almacenamiento guarda las matrices generadas, los identificadores para cada matriz (`names.mat`) y una serie de gráficas que muestran en un plano la forma de las evaluaciones, éstas contienen su identificador para que pueden visualizarse.

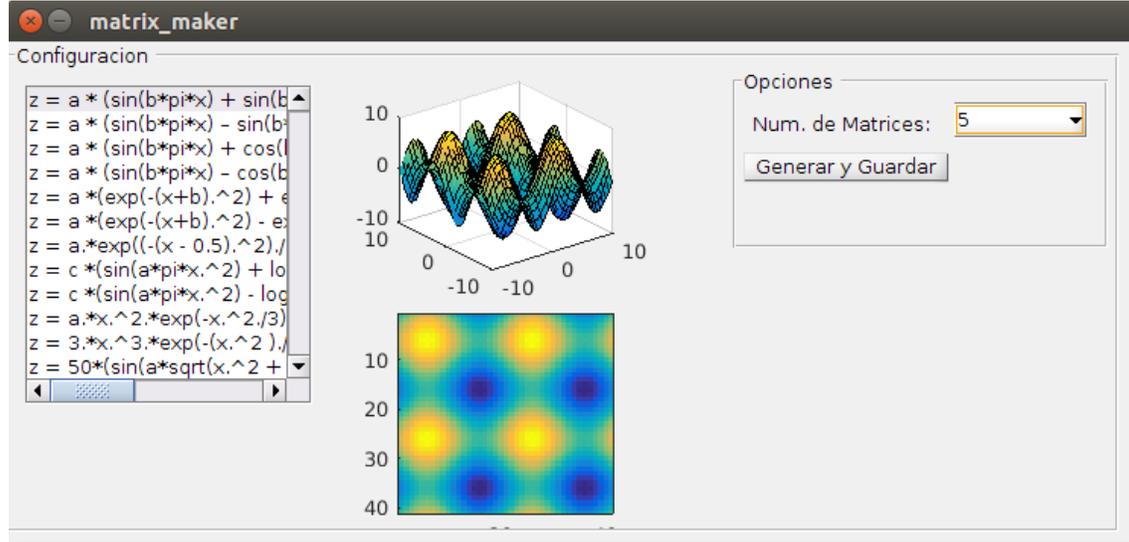


Figura D.1: Interfaz principal de la herramienta Matrix Maker.

D.0.3. Funcionamiento técnico

La herramienta sigue el siguiente esquema funcional, donde se muestran los componentes de la herramienta, ver la Fig. D.3.

Se genera archivos con extensión `.mat`. Las matrices mantienen la siguiente convención: `ID'xx.mat`, donde `xx` es un numero secuencial. La forma de cargar esta variable en el espacio de trabajo de matlab es la siguiente:

```
1 >> objeto = load('ID.12.mat')
2 >> matriz = objeto.M
```

El archivo `names.mat`, contiene una variable de Matlab que se llama `namesAdded`, se puede obtener fácilmente utilizando el siguiente codigo:

```
1 >> matriz = load('names.mat')
2 >> map = matriz.namesAdded
3 >> map('ID_1')
4     ans =
5
6     $-0.9 (\sin(-0.9 \pi x)+\sin(-0.9 \pi x))$
```

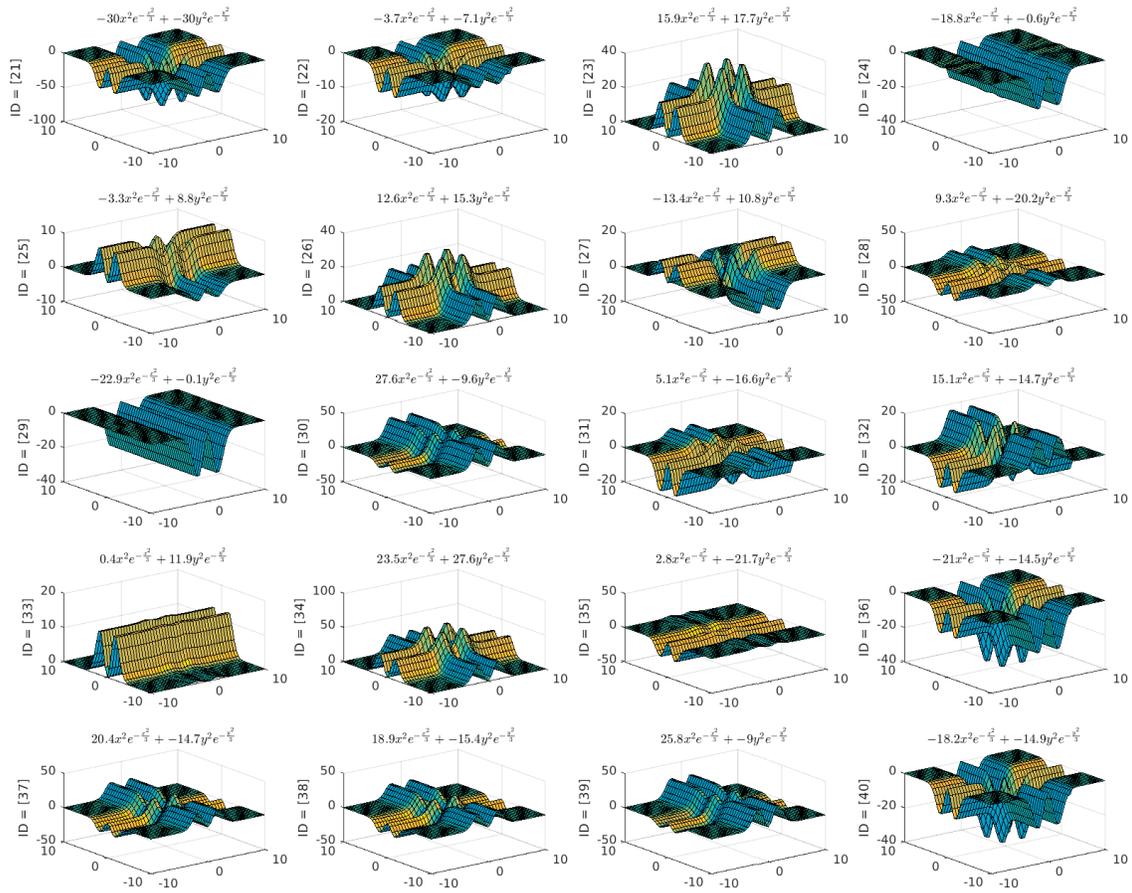


Figura D.2: Vista de algunas matrices generadas, notese que se muestra la función, con su respectivo ID y la etiqueta de la función con sus parámetros.

Cada identificador de la matriz contiene una cadena de texto formateada en ecuaciones de latex. Allí se muestran los parámetros con los que fue evaluada la función.

D.1. Código fuente

Listing D.1: `matrixmaker.m`

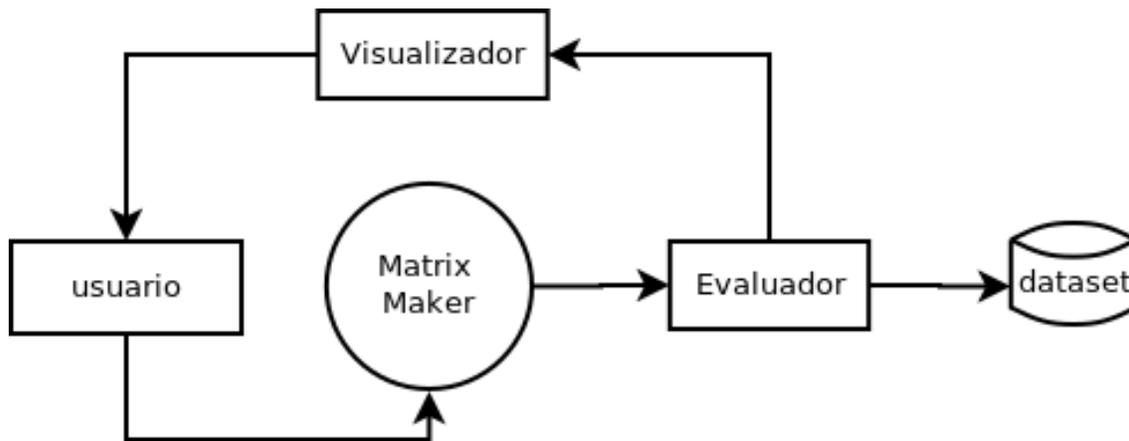


Figura D.3: Diagrama de contexto del funcionamiento de la herramienta.

```

1 function varargout = matrix_maker(varargin)
2 %Begin initialization code – DO NOT EDIT
3 gui_Singleton = 1;
4 gui_State = struct('gui_Name',       mfilename, ...
5                   'gui_Singleton',  gui_Singleton, ...
6                   'gui_OpeningFcn', @matrix_maker_OpeningFcn, ...
7                   'gui_OutputFcn',  @matrix_maker_OutputFcn, ...
8                   'gui_LayoutFcn',  [], ...
9                   'gui_Callback',    []);
10 if nargin && ischar(varargin{1})
11     gui_State.gui_Callback = str2func(varargin{1});
12 end
13
14 if nargin
15     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
16 else
17     gui_mainfcn(gui_State, varargin{:});
18 end
19
20 function matrix_maker_OpeningFcn(hObject, eventdata, handles, ...
21     varargin)
22 names = containers.Map ;
23 handles.output = hObject;
24 handles.contID = 1;
25 handles.names = names;
26 guidata(hObject, handles);
27
28 function varargout = matrix_maker_OutputFcn(hObject, eventdata, ...
29     handles)
30 varargout{1} = handles.output;
31
32 function listbox1_Callback(hObject, eventdata, handles)
  
```

```

31     index_selected = get(hObject, 'Value');
32     list = get(hObject, 'String');
33     item_selected = list{index_selected};
34
35     [x,y,z] = evaluarFuncion(index_selected);
36
37     surf(handles.axes1,x,y,z);
38     title('Muestra');
39     shading flat
40     shading interp
41     rotate3d on
42     imagesc(handles.axes4,z);
43     handles.index_selected = index_selected;
44     guidata(hObject, handles);
45
46 function listBox1_CreateFcn(hObject, eventdata, handles)
47 if ispc && isequal(get(hObject, 'BackgroundColor'), ...
48     get(0, 'defaultUicontrolBackgroundColor'))
49     set(hObject, 'BackgroundColor', 'white');
50 end
51 function pushbutton1_Callback(hObject, eventdata, handles)
52     names = handles.names;
53     contenido = get(handles.popupmenu1, 'String');
54     contID = handles.contID;
55     nMtrx = str2num(contenido{get(handles.popupmenu1, 'Value')});
56     dir = mfilename('fullpath');
57     currentPath = fileparts(dir);
58     dname = uigetdir(currentPath);
59
60     %se puede cambiar aqui el evaluador: evaluador -> ...
61     evaluador_aleatorio
62     [contID, namesAdded]= ...
63     evaluador_aleatorio(handles.index_selected, nMtrx, dname, ...
64     contID, names);
65
66     %CREAR ARCHIVO names.MAT
67     nombresFile = strcat(dname, '/names.mat');
68     save(nombresFile, 'namesAdded');
69     handles.names = namesAdded;
70     handles.contID = contID;
71     guidata(hObject, handles);
72
73 function edit2_Callback(hObject, eventdata, handles)
74 function edit2_CreateFcn(hObject, eventdata, handles)

```

```

75 if ispc && isequal(get(hObject, 'BackgroundColor'), ...
    get(0, 'defaultUicontrolBackgroundColor'))
76     set(hObject, 'BackgroundColor', 'white');
77 end
78
79 function pushbutton4_Callback(hObject, eventdata, handles)
80
81 function pushbutton2_Callback(hObject, eventdata, handles)
82
83 function popupmenu1_Callback(hObject, eventdata, handles)
84
85 function popupmenu1_CreateFcn(hObject, eventdata, handles)
86 if ispc && isequal(get(hObject, 'BackgroundColor'), ...
    get(0, 'defaultUicontrolBackgroundColor'))
87     set(hObject, 'BackgroundColor', 'white');
88 end
89
90 function edit3_Callback(hObject, eventdata, handles)
91
92 function edit3_CreateFcn(hObject, eventdata, handles)
93 if ispc && isequal(get(hObject, 'BackgroundColor'), ...
    get(0, 'defaultUicontrolBackgroundColor'))
94     set(hObject, 'BackgroundColor', 'white');
95 end
96
97 function edit4_Callback(hObject, eventdata, handles)
98
99 function edit4_CreateFcn(hObject, eventdata, handles)
100 if ispc && isequal(get(hObject, 'BackgroundColor'), ...
    get(0, 'defaultUicontrolBackgroundColor'))
101     set(hObject, 'BackgroundColor', 'white');
102 end

```

Listing D.2: evaluador.m

```

1  % *****
2  % Funcion escrita por
3  % Angel Refugio Mandujano Garcia
4  % %ultima revision: nov 15 de 2016
5  % a.mandujanogarcia@gmail.com
6  % *****
7  % Funcion que evalua de manera estatica las funciones predefinidas.
8  % Se ocupa para generar las matrices de datos sinteticos de ...
    manera SECUENCIAL, guarda la informacion en un directorio dado.
9  % y guarda las variables necesarias en archivos .mat
10 %
11 % ENTRADAS

```

```

12 %index_func = indice de la funcion (1-12)
13 %nFuncs = numero de evaluaciones que se requieren
14 %folder = directorio donde se guardaran las matrices generadas
15 %contID = contador de las matrices
16 %names = variable de los Identificadores de cada matriz
17 %
18 %
19 %SALIDAS
20 %contID = contador de las matrices generadas (por si se ...
    requiere evaluar mas de una funcion)
21 %namesAdded = variable de los Identificadores de cada matriz ...
    generada (por si se requiere evaluar mas de una funcion)
22 function [contID,namesAdded] = evaluador(index_func , ...
    nFuncs , folder , contID , names)
23
24 %CREAR Y VISUALIZAR LAS MATRICES
25 [x,y]=meshgrid(-10:.5:10);
26 H = Funciones();
27
28 [P,Q] = size(x);
29 num_decimales = 1;
30 a = 0;
31 b = 0;
32 c = 0;
33 contID_ini = contID;
34 figure1 = figure('units','normalized','outerposition',[0 0 1 ...
    1]);
35 for i=1:nFuncs
36     subplot(5,4,i);
37     if index_func == 1
38         if i == 1; a = -0.9 ;b = -0.9; end %inicializa ...
            cuando es primera matriz
39         titulo = strcat('$',num2str(a),' ...
            (\sin(',num2str(b),' \pi x)+\sin(',num2str(b),' ...
            \pi x))$');
40         for m = 1:P
41             for n = 1:Q
42                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
43             end
44         end
45         a = a + 0.1;
46         b = b + 0.1;
47
48     elseif index_func == 2
49         if i == 1; a = -0.9 ;b = -0.9; end %inicializa ...
            cuando es primera matriz
50         titulo = strcat('$',num2str(a),' ...
            (\sin(',num2str(b),' \pi x)-\sin(',num2str(b),' ...
            \pi x))$');

```

```

51         for m = 1:P
52             for n = 1:Q
53                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
54             end
55         end
56         a = a + 0.1;
57         b = b + 0.1;
58
59     elseif index_func == 3
60         if i == 1; a = -0.9 ;b = -0.9; end % inicializa ...
           cuando es primera matriz
61         titulo = strcat('$',num2str(a),' ...
           (\sin(' ,num2str(b), ' \pi x)+\cos(' ,num2str(b), ' ...
           \pi x))$');
62         for m = 1:P
63             for n = 1:Q
64                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
65             end
66         end
67         a = a + 0.1;
68         b = b + 0.1;
69
70     elseif index_func == 4
71         if i == 1; a = -0.9 ;b = -0.9; end % inicializa ...
           cuando es primera matriz
72         titulo = strcat('$',num2str(a),' ...
           (\sin(' ,num2str(b), ' \pi x)-\cos(' ,num2str(b), ' ...
           \pi x))$');
73         for m = 1:P
74             for n = 1:Q
75                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
76             end
77         end
78         a = a + 0.1;
79         b = b + 0.1;
80
81     elseif index_func == 5
82         if i == 1; a = -8; b = -8; end % inicializa cuando ...
           es primera matriz
83         titulo = strcat('$',num2str(a),' ...
           (e^{-(x+' ,num2str(b), ')^2}+e^{-(y-' ,num2str(b), ')^2})$');
84         for m = 1:P
85             for n = 1:Q
86                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
87             end
88         end
89         a = a + 1;
90         b = b + 1;
91

```

```

92     elseif index_func == 6
93         if i == 1; a = -20 ;b = -7; end % inicializa cuando ...
           es primera matriz
94         titulo = strcat('$',num2str(a),' ...
           (e^{-x+}',num2str(b),' )^{2}}-e^{-y-',num2str(b),' )^{2}})$');
95         for m = 1:P
96             for n = 1:Q
97                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
98             end
99         end
100        a = a + 2;
101        b = b + 1;
102
103    elseif index_func == 7
104        if i == 1; a = -15 ;b = -7; end % inicializa cuando ...
           es primera matriz
105        titulo = strcat('$',num2str(a),' ...
           e^{-\frac{(x-0.5)^2}{15}}+',num2str(b),' ...
           e^{-\frac{(y-0.5)^2}{15}})$');
106        for m = 1:P
107            for n = 1:Q
108                M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
109            end
110        end
111        a = a + 2;
112        b = b + 1;
113
114    elseif index_func == 8
115        if i == 1; a = -10 ;b = 1; c = 1; end % inicializa ...
           cuando es primera matriz
116        titulo = strcat('$',num2str(c),' \sin(',num2str(a),' ...
           \pi x^2)+\log(|',num2str(b),' \pi y|)$');
117        for m = 1:P
118            for n = 1:Q
119                M(m,n) = H{index_func}(x(m,n),y(m,n),a,b,c);
120            end
121        end
122        a = a + 1;
123        b = b + 1;
124
125    elseif index_func == 9
126        if i == 1; a = -10 ;b = 1; c = -1;end % inicializa ...
           cuando es primera matriz
127        titulo = strcat('$',num2str(c),' \sin(',num2str(a),' ...
           \pi x^2)- \log(|',num2str(b),' \pi y|)$');
128        for m = 1:P
129            for n = 1:Q
130                M(m,n) = H{index_func}(x(m,n),y(m,n),a,b,c);
131        end

```

```

132         end
133         a = a + 1;
134         b = b + 1;
135
136     elseif index_func == 10
137         if i == 1; a = -30 ;b = -30; end % inicializa cuando ...
            es primera matriz
138         titulo = strcat('$', num2str(a), ' x^{2} ...
            e^{-\frac{x^{2}}{3}} + ', num2str(b), ' y^{2} ...
            e^{-\frac{y^{2}}{3}} $');
139
140         for m = 1:P
141             for n = 1:Q
142                 M(m,n) = H{index_func}(x(m,n), y(m,n), a, b);
143             end
144         end
145         a = a + 3;
146         b = b + 5;
147
148     elseif index_func == 11
149         if i == 1; a = -30 ;b = -30; end % inicializa cuando ...
            es primera matriz
150         titulo = strcat('$', num2str(a), ' x^{3} ...
            e^{-\frac{x^{2}}{3}} + ', num2str(b), ' y^{3} ...
            e^{-\frac{y^{2}}{3}} $');
151         for m = 1:P
152             for n = 1:Q
153                 M(m,n) = H{index_func}(x(m,n), y(m,n), a, b);
154             end
155         end
156         a = a + 3;
157         b = b + 3;
158
159     elseif index_func == 12
160         if i == 1; a = -10 ;b = -10; c = -10; end % ...
            inicializa cuando es primera matriz
161         titulo = strcat('$50 (\frac{\sin(' , num2str(a), ' ...
            \sqrt{x^{2}+y^{2}} + ', num2str(c), ' )', num2str(b), ' ...
            \sqrt{x^{2}+y^{2}} + ', num2str(c), ' ) $');
162         for m = 1:P
163             for n = 1:Q
164                 M(m,n) = H{index_func}(x(m,n), y(m,n), a, b, c);
165             end
166         end
167         a = a + 2;
168         b = b + 2;
169         c = c + 2;
170
171     end

```

```

172
173     surfl(x,y,M)
174     rotate3d on
175     title(titulo, 'Interpreter', 'latex', 'fontsize', 8);
176     zlbl = strcat('ID = [', num2str(contID), ']');
177     xlabel(zlbl, 'fontsize', 8);
178     xt = get(gca, 'XTick');
179     set(gca, 'FontSize', 8)
180
181     %CREAR ARCHIVO .MAT de la matriz
182     stringID = strcat('ID_', num2str(contID));
183     fileMat = strcat(folder, '/', stringID, '.mat');
184     save(fileMat, 'M');
185
186     %CREAR LISTAS PARA VARIABLE: names
187     nvar = strcat('ID_', num2str(contID));
188     names(nvar) = titulo;
189
190     contID = contID + 1;
191 end
192
193 %CREAR IMAGEN DE MUESTRAS
194 fileplot1 = ...
195     strcat(folder, '/file_', num2str(contID_ini), '-', num2str(contID-1), '.png');
196 saveas(figure1, fileplot1);
197 namesAdded = names;
198 end
199 function [H] = Funciones()
200 H{1} = @(x,y,a,b) a * (sin(b*pi*x) + sin(b*pi*y));
201 H{2} = @(x,y,a,b) a * (sin(b*pi*x) - sin(b*pi*y));
202 H{3} = @(x,y,a,b) a * (sin(b*pi*x) + cos(b*pi*y));
203 H{4} = @(x,y,a,b) a * (sin(b*pi*x) - cos(b*pi*y));
204 H{5} = @(x,y,a,b) a * (exp(-(x+b).^2) + exp(-(y-b).^2));
205 H{6} = @(x,y,a,b) a * (exp(-(x+b).^2) - exp(-(y-b).^2));
206 H{7} = @(x,y,a,b) a.*exp((-x - 0.5).^2 ./15) + b.*exp((-y ...
    - 0.5).^2 ./15);
207 H{8} = @(x,y,a,b,c) c * (sin(a*pi*x.^2) + log(abs(b*pi*y)));
208 H{9} = @(x,y,a,b,c) c * (sin(a*pi*x.^2) - log(abs(b*pi*y)));
209 H{10} = @(x,y,a,b) a.*x.^2.*exp(-x.^2./3) + ...
    b.*y.^2.*exp(-y.^2./3);
210 H{11} = @(x,y,a,b) a.*x.^3.*exp(-x.^2 ./3) + ...
    b.*y.^3.*exp(-y.^2 ./3);
211 H{12} = @(x,y,a,b,c) 50*(sin(a*sqrt(x.^2 + y.^2) + c) ./ ...
    (b*sqrt(x.^2 + y.^2) + c));
212 end

```

Listing D.3: *evaluadoraleatorio.m*

```

1  % *****
2  % Funcion escrita por
3  % Angel Refugio Mandujano Garcia
4  %%ultima revision: nov 15 de 2016
5  % a.mandujanogarcia@gmail.com
6  % *****
7  % Funcion que evalua de manera estatica las funciones predefinidas.
8  % Se ocupa para generar las matrices de datos sinteticos de ...
   % manera ALEATORIA, guarda la informacion en un directorio dado.
9  % y guarda las variables necesarias en archivos .mat
10 %
11 %ENTRADAS
12 %index_func = indice de la funcion (1-12)
13 %nFuncs = numero de evaluaciones que se requieren
14 %folder = directorio donde se guardaran las matrices generadas
15 %contID = contador de las matrices
16 %names = variable de los Identificadores de cada matriz
17 %
18 %SALIDAS
19 %contID = contador de las matrices generadas (por si se ...
   % requiere evaluar mas de una funcion)
20 %namesAdded = variable de los Identificadores de cada matriz ...
   % generada (por si se requiere evaluar mas de una funcion)
21
22 function [contID,namesAdded] = evaluador_aleatorio(index_func, ...
   nFuncs, folder, contID, names)
23     [x,y]=meshgrid(-10:.5:10);
24     H = Funciones();
25
26     [P,Q] = size(x);
27     num_decimales = 1;
28     a = 0;
29     b = 0;
30     c = 0;
31     contID_ini = contID;
32     figure1 = figure('units','normalized','outerposition',[0 0 1 ...
   1]);
33     for i=1:nFuncs
34         subplot(5,4,i);
35         if index_func == 1
36             if i == 1; a = -0.9 ;b = -0.9; end % inicializa ...
   % cuando es primera matriz
37             titulo = strcat('$',num2str(a),' ...
   (\sin(' ,num2str(b), ' \pi x)+\sin(' ,num2str(b), ' ...
   \pi x))$');
38             for m = 1:P
39                 for n = 1:Q
40                     M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);

```

```

41         end
42     end
43     a = -0.9 + (0.9 - (-0.9))*rand(1,1);
44     a = round(a*(10^num_decimales))/(10^num_decimales);
45     b = -0.9 + (0.9 - (-0.9))*rand(1,1);
46     b = round(b*(10^num_decimales))/(10^num_decimales);
47
48
49 elseif index_func == 2
50     if i == 1; a = -0.9 ;b = -0.9; end % inicializa ...
        cuando es primera matriz
51     titulo = strcat('$',num2str(a),' ...
        (\sin(',num2str(b),' \pi x)-\sin(',num2str(b),' ...
        \pi x))$');
52     for m = 1:P
53         for n = 1:Q
54             M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
55         end
56     end
57     a = -0.9 + (0.9 - (-0.9))*rand(1,1);
58     a = round(a*(10^num_decimales))/(10^num_decimales);
59     b = -0.9 + (0.9 - (-0.9))*rand(1,1);
60     b = round(b*(10^num_decimales))/(10^num_decimales);
61
62 elseif index_func == 3
63     if i == 1; a = -0.9 ;b = -0.9; end % inicializa ...
        cuando es primera matriz
64     titulo = strcat('$',num2str(a),' ...
        (\sin(',num2str(b),' \pi x)+\cos(',num2str(b),' ...
        \pi x))$');
65     for m = 1:P
66         for n = 1:Q
67             M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
68         end
69     end
70     a = -0.9 + (0.9 - (-0.9))*rand(1,1);
71     a = round(a*(10^num_decimales))/(10^num_decimales);
72     b = -0.9 + (0.9 - (-0.9))*rand(1,1);
73     b = round(b*(10^num_decimales))/(10^num_decimales);
74
75 elseif index_func == 4
76     if i == 1; a = -0.9 ;b = -0.9; end % inicializa ...
        cuando es primera matriz
77     titulo = strcat('$',num2str(a),' ...
        (\sin(',num2str(b),' \pi x)-\cos(',num2str(b),' ...
        \pi x))$');
78     for m = 1:P
79         for n = 1:Q
80             M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);

```

```

81         end
82     end
83     a = -0.9 + (0.9 - (-0.9))*rand(1,1);
84     a = round(a*(10^num_decimales))/(10^num_decimales);
85     b = -0.9 + (0.9 - (-0.9))*rand(1,1);
86     b = round(b*(10^num_decimales))/(10^num_decimales);
87
88     elseif index_func == 5
89         if i == 1; a = -8; b = -8; end % inicializa cuando ...
90         es primera matriz
91         titulo = strcat('$', num2str(a), ' ...
92             (e^{-(x+', num2str(b), ')^2} + e^{-(y-', num2str(b), ')^2})$');
93         for m = 1:P
94             for n = 1:Q
95                 M(m,n) = H{index_func}(x(m,n), y(m,n), a, b);
96             end
97         end
98         a = -15 + (15 - (-15))*rand(1,1);
99         a = round(a*(10^num_decimales))/(10^num_decimales);
100        b = -7 + (7 - (-7))*rand(1,1);
101        b = round(b*(10^num_decimales))/(10^num_decimales);
102
103    elseif index_func == 6
104        if i == 1; a = -20 ; b = -7; end % inicializa cuando ...
105        es primera matriz
106        titulo = strcat('$', num2str(a), ' ...
107            (e^{-(x+', num2str(b), ')^2} - e^{-(y-', num2str(b), ')^2})$');
108        for m = 1:P
109            for n = 1:Q
110                M(m,n) = H{index_func}(x(m,n), y(m,n), a, b);
111            end
112        end
113        a = 1 + (20 - (-20))*rand(1,1);
114        a = round(a*(10^num_decimales))/(10^num_decimales);
115        b = -7 + (7 - (-7))*rand(1,1);
116        b = round(b*(10^num_decimales))/(10^num_decimales);
117
118    elseif index_func == 7
119        if i == 1; a = -15 ; b = -7; end % inicializa cuando ...
120        es primera matriz
121        titulo = strcat('$', num2str(a), ' ...
122            e^{-\frac{(x-0.5)^2}{15}} + ', num2str(b), ' ...
123            e^{-\frac{(y-0.5)^2}{15}}$');
124        for m = 1:P
125            for n = 1:Q
126                M(m,n) = H{index_func}(x(m,n), y(m,n), a, b);
127            end
128        end
129        a = -15 + (15 - (-15))*rand(1,1);

```

```

123         a = round(a*(10^num_decimales))/(10^num_decimales);
124         b = -7 + (7 - (-7))*rand(1,1);
125         b = round(b*(10^num_decimales))/(10^num_decimales);
126
127     elseif index_func == 8
128         if i == 1; a = -10 ;b = 1; c = 1; end % inicializa ...
           cuando es primera matriz
129         titulo = strcat('$',num2str(c), ' \sin(' ,num2str(a), ' ...
           \pi x^{2})+\log(|',num2str(b), ' \pi y|)$');
130         for m = 1:P
131             for n = 1:Q
132                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b,c);
133             end
134         end
135         a = -10 + (10 - (-10))*rand(1,1);
136         a = round(a*(10^num_decimales))/(10^num_decimales);
137         b = -1 + (15 - (-1))*rand(1,1);
138         b = round(b*(10^num_decimales))/(10^num_decimales);
139
140     elseif index_func == 9
141         if i == 1; a = -10 ;b = 1; c = -1;end % inicializa ...
           cuando es primera matriz
142         titulo = strcat('$',num2str(c), ' \sin(' ,num2str(a), ' ...
           \pi x^{2})-\log(|',num2str(b), '\pi y|)$');
143         for m = 1:P
144             for n = 1:Q
145                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b,c);
146             end
147         end
148         a = -10 + (10 - (-10))*rand(1,1);
149         a = round(a*(10^num_decimales))/(10^num_decimales);
150         b = -1 + (15 - (-1))*rand(1,1);
151         b = round(b*(10^num_decimales))/(10^num_decimales);
152
153     elseif index_func == 10
154         if i == 1; a = -30 ;b = -30; end % inicializa cuando ...
           es primera matriz
155         titulo = strcat('$',num2str(a), ' x^{2} ...
           e^{-\frac{x^{2}}{3}}+',num2str(b), ' y^{2} ...
           e^{-\frac{y^{2}}{3}}$');
156
157         for m = 1:P
158             for n = 1:Q
159                 M(m,n) = H{index_func}(x(m,n),y(m,n),a,b);
160             end
161         end
162         a = -30 + (30 - (-30))*rand(1,1);
163         a = round(a*(10^num_decimales))/(10^num_decimales);
164         b = -30 + (30 - (-30))*rand(1,1);

```

```

165         b = round(b*(10^ num_decimales))/(10^ num_decimales);
166
167     elseif index_func == 11
168         if i == 1; a = -30 ;b = -30; end % inicializa cuando ...
            es primera matriz
169         titulo = strcat('$', num2str(a), ' x^{3} ...
            e^{-\frac{x^2}{3}}+ ', num2str(b), ' y^{3} ...
            e^{-\frac{y^2}{3}}$');
170         for m = 1:P
171             for n = 1:Q
172                 M(m,n) = H{index_func}(x(m,n), y(m,n), a, b);
173             end
174         end
175         a = -30 + (30 - (-30))*rand(1,1);
176         a = round(a*(10^ num_decimales))/(10^ num_decimales);
177         b = -30 + (30 - (-30))*rand(1,1);
178         b = round(b*(10^ num_decimales))/(10^ num_decimales);
179
180     elseif index_func == 12
181         if i == 1; a = -10 ;b = -10; c = -10; end % ...
            inicializa cuando es primera matriz
182         titulo = strcat('$50 (\frac{\sin(', num2str(a), ' ...
            \sqrt{x^2+y^2}}+ ', num2str(c), ' )', num2str(b), ' ...
            \sqrt{x^2+y^2}}+ ', num2str(c), ' )$');
183         for m = 1:P
184             for n = 1:Q
185                 M(m,n) = H{index_func}(x(m,n), y(m,n), a, b, c);
186             end
187         end
188         a = -10 + (10 - (-10))*rand(1,1);
189         a = round(a*(10^ num_decimales))/(10^ num_decimales);
190         b = -10 + (10 - (-10))*rand(1,1);
191         b = round(b*(10^ num_decimales))/(10^ num_decimales);
192         c = -10 + (10 - (-10))*rand(1,1);
193         c = round(c*(10^ num_decimales))/(10^ num_decimales);
194     end
195
196     surf(x,y,M)
197     rotate3d on
198     title(titulo, 'Interpreter', 'latex', 'fontSize', 8);
199     zlbl = strcat('ID = [', num2str(contID), ']');
200     xlabel(zlbl, 'fontSize', 8);
201     xt = get(gca, 'XTick');
202     set(gca, 'FontSize', 8)
203
204     %CREAR ARCHIVO .MAT de la matriz
205     stringID = strcat('ID_', num2str(contID));
206     fileMat = strcat(folder, '/', stringID, '.mat');
207     save(fileMat, 'M');

```

```

208
209         %CREAR LISTAS PARA VARIABLE: names
210         nvar = strcat('ID_', num2str(contID));
211         names(nvar) = titulo;
212         contID = contID + 1;
213     end
214
215     %CREAR IMAGEN DE MUESTRAS
216     fileplot1 = ...
                strcat(folder, 'file_', num2str(contID_ini), '-', num2str(contID-1), '.png');
217     saveas(figure1, fileplot1);
218     namesAdded = names;
219 end
220
221 function [H] = Funciones()
222     H{1} = @(x,y,a,b) a * (sin(b*pi*x) + sin(b*pi*y));
223     H{2} = @(x,y,a,b) a * (sin(b*pi*x) - sin(b*pi*y));
224     H{3} = @(x,y,a,b) a * (sin(b*pi*x) + cos(b*pi*y));
225     H{4} = @(x,y,a,b) a * (sin(b*pi*x) - cos(b*pi*y));
226     H{5} = @(x,y,a,b) a * (exp(-(x+b).^2) + exp(-(y-b).^2));
227     H{6} = @(x,y,a,b) a * (exp(-(x+b).^2) - exp(-(y-b).^2));
228     H{7} = @(x,y,a,b) a.*exp((-x - 0.5).^2 ./15) + b.*exp((-y ...
                - 0.5).^2 ./15);
229     H{8} = @(x,y,a,b,c) c * (sin(a*pi*x.^2) + log(abs(b*pi*y)));
230     H{9} = @(x,y,a,b,c) c * (sin(a*pi*x.^2) - log(abs(b*pi*y)));
231     H{10} = @(x,y,a,b) a.*x.^2.*exp(-x.^2./3) + ...
                b.*y.^2.*exp(-y.^2./3);
232     H{11} = @(x,y,a,b) a.*x.^3.*exp(-x.^2 ./3) + ...
                b.*y.^3.*exp(-y.^2 ./3);
233     H{12} = @(x,y,a,b,c) 50*(sin(a*sqrt(x.^2 + y.^2) + c) ./ ...
                (b*sqrt(x.^2 + y.^2) + c));
234 end

```

Listing D.4: evaluarFuncion.m

```

1  % *****
2  % Funcion escrita por
3  % Angel Refugio Mandujano Garcia
4  % %ultima revision: nov 15 de 2016
5  % a.mandujanogarcia@gmail.com
6  % *****
7  % Funcion que evalua de manera estatica las funciones ...
    predefinidas. Para la visualizacion de ejemplo en la ...
    interfaz principal
8  %
9  % ENTRADAS
10 % index_func = indice de la funcion (1-12)

```

```

11 %
12 %SALIDAS
13 %x = valores en el eje x (-10:.5:10)
14 %y = valores en el eje y (-10:.5:10)
15 %z = valores en el eje z, corresponden a la evaluacion de la matriz
16 %
17 function [x,y,z] = evaluarFuncion(index_func)
18     [x,y]=meshgrid(-10:.5:10);
19
20     if index_func == 1
21         z = 5 * (sin(0.2*pi*x) + sin(0.2*pi*y));
22     elseif index_func == 2
23         z = 5 * (sin(0.2*pi*x) - sin(0.2*pi*y));
24     elseif index_func == 3
25         z = 5 * (sin(0.23*pi*x) + cos(0.2*pi*y));
26     elseif index_func == 4
27         z = 5 * (sin(0.2*pi*x) - cos(0.2*pi*y));
28     elseif index_func == 5
29         z = -5 * (exp(-(x+0.5).^2) + exp(-(y-0.5).^2));
30     elseif index_func == 6
31         z = 5 * (exp(-(x+0.5).^2) - exp(-(y-0.5).^2));
32     elseif index_func == 7
33         z = 3.*exp((-x - 0.5).^2)./15) + 5.*exp(-(y - ...
34             0.5).^2)./15;
35     elseif index_func == 8
36         z = 1 * (sin(3*pi*x.^2) + log(abs(6*pi*y)));
37     elseif index_func == 9
38         z = 1 * (sin(3*pi*x.^2) - log(abs(6*pi*y)));
39     elseif index_func == 10
40         z = 8.*x.^2.*exp(-x.^2./3) + 8.*y.^2.*exp(-y.^2./3);
41     elseif index_func == 11
42         z = 3.*x.^3.*exp(-(x.^2 )./3) + 3.*y.^3.*exp(-(y.^2 )./3);
43     elseif index_func == 12
44         z = 1*(sin(1*sqrt(x.^2 + y.^2 ) + eps)./ (1*sqrt(x.^2 + ...
45             y.^2) + eps));
46     end
47 end

```

Apéndice E

Script de validación

Listing E.1: Dendograma.m

```
1  % *****
2  % Funcion escrita por
3  % Angel Refugio Mandujano Garcia
4  % ultima revision: nov 15 de 2016
5  % a.mandujanogarcia@gmail.com
6  % *****
7  % * Funcion que realiza el las comparaciones de todos los ...
   elementos contra
8  % todos en un directorio.
9  % * Se obtiene la matriz de distancias para los elementos
10 % * Mediante un analisis de recuperacion se obtienen un indice de
11 % recuperacion para cada elemento a su categoria
12 %
13
14 function [zero] = Dendograma()
15     indexTecnica = 10;
16     disp(strcat('tecnica: ',int2str(indexTecnica)));
17     pathc = mfilename('fullpath'); %%obtiene el directorio y ...
       script actual
18     [currentpath, filenames, fileextension] = fileparts(pathc); ...
       %%separa los
19     addpath(strcat(currentpath, '/tecnicas')); %%agrega los ...
       subdirectorios
20     fp = fileparts(currentpath);
21
22     directory = uigetdir(fp, 'Buscar');
23     allFiles = dir(directory);
24     fileList = {allFiles(~[allFiles.isdir]).name};
25
26     % lista de los elementos del directorio
```

```

27     c = length(fileList);
28
29     % se le asigna un identificador a cada elemento y se inicializa.
30     for i = 1: c
31         ruta = strcat(directory, '/', char(fileList(i)));
32         nombres{i} = strcat('obj', int2str(i));
33         img = imread(ruta);
34         imgBrutas{i} = img;
35         img = inicializaImage(img);
36         matrices{i} = img;
37     end
38
39     % se guarda la informacion en un map(key, matriz)
40     objetos = containers.Map(nombres, matrices); %
41     objetosBruto = containers.Map(nombres, imgBrutas);
42
43
44     % Se realiza la matriz de distancias. se realizan (25*24)/2
45     % comparaciones
46     valores = [];
47     A = 1;
48     B = 2;
49     tic;
50     while A ≤ c
51         id_X = strcat('obj', int2str(A));
52         X = objetos(id_X);
53         b = B;
54         while b ≤ c
55             id_Y = strcat('obj', int2str(b));
56             Y = objetos(id_Y);
57             n = Tecnicas(indexTecnica, double(X), double(Y));
58             valores = [valores n];
59             b = b + 1;
60         end
61         A = A + 1;
62         B = B + 1;
63     end
64
65     toc
66
67     % dado el arreglo de valores de la comparacion, se crea la ...
68     matriz de
69     % distancias.
70     MD = squareform(valores);
71     %tree = linkage(MD);
72     %figure()
73     %dendrogram(tree);
74

```

```
75     % se obtienen los identificadores de cada elemento y se ...
       guardan en una
76     % matriz de objetos.
77     MObjs = {}
78     [Vsorted, SortIndex] = sort(MD);
79     for ii=1 :c
80         a = SortIndex(:, ii);
81         objNamsSorted = nombres(a);
82         objTagsSorted = fileList(a);
83         MObjs(:, ii) = cellstr(objNamsSorted');
84     end
85
86     disp(MObjs);
87     % guardar variables
88     fileExp = strcat(currentpath, '/Resultados/valores.mat');
89     save(fileExp, 'valores');
90     zero = 0;
91 end
```

Apéndice F

Matrices Artificiales I

Este apartado se refiere a la creación de matrices de datos creados artificialmente. El nombre del apartado lleva un numero (I), significa que es la primera parte de los datos que se crearan artificialmente.

Estos datos artificiales corresponden a la forma mas sencilla de crearlos, esto es, se han creado nueve matrices de datos tan solo moviendo una ventana de datos que van recorriendo la matriz en diferentes posiciones como se muestra en la Fig. F.1.

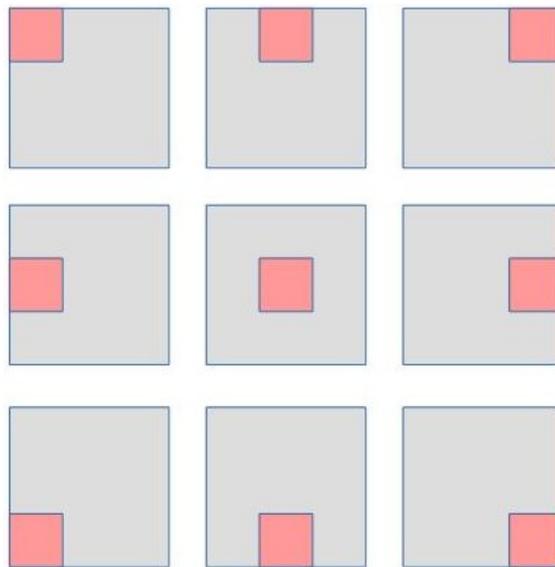


Figura F.1: Organización básica de las matrices artificiales

Esta herramienta se encuentra ubicada en la siguiente dirección del toolbox de

matlab desarrollado:

```
../COMPARACION MATRICIAL/CODIGO FUENTE/MISC
```

El archivo se llama: `matrices_a_mano.m` y debe ser ejecutado antes de ejecutar la aplicación completa, para así tener las matrices de prueba necesarias.

Este archivo recibe como parámetros tres números enteros:

N = tamaño de las matrices (solo matrices cuadradas).

R = número que servirá como contenido de la matriz.

W = número que servirá para llenar la submatriz de ventana.

De esta manera `matrices_a_mano.m` creará una matriz de tamaño 5×5 , llena con unos pero con 5 en la ventana:

```
5 5 1 1 1
5 5 1 1 1
1 1 1 1 1
1 1 1 1 1
```

Lo que ocurre dentro de este archivo es mover una ventana conteniendo algún número dentro de una matriz de unos y guardar esa matriz en formato `.mat` para su uso posterior.

A continuación se muestran algunas de las salidas con diferentes parámetros de entrada:

F.1. Código Fuente

Listing F.1: matrices_a_mano.m

```
1 %esta funcion crea 9 matrices llenas de ceros , excepto una ...
   ventana de unos que se mueve a lo largo de la matriz.
2 %cada matriz creada corresponde al desplazamiento de una ...
   ventana de unos.
3 %N = tamaño de las matrices que se crearan
4 %C = contenido de la matriz
5 %W = contenido de la ventana
```

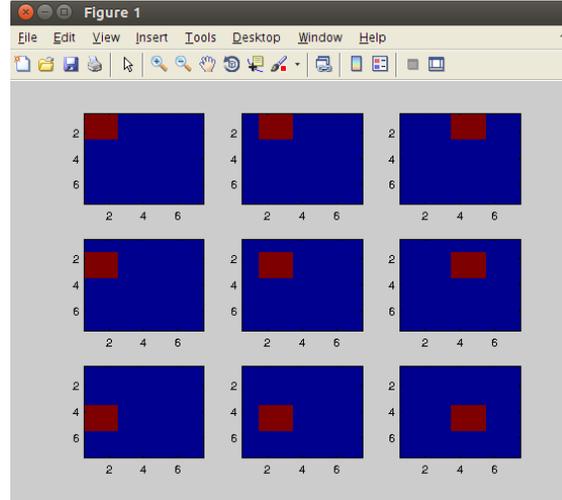


Figura F.2: Matriz de 7×7 llena de ceros con ventana de unos.

```

6 function matrices = matrices_a_mano(N,C,W)
7     window = round(N/3); % tamaño de la ventana
8     a = [1 window (window*2)] % posiciones de la ventana dentro ...
        de la matriz N
9     matriz = C*ones(N,N); % matriz auxiliar
10    cont = 1;
11    M = []; %
12    dirDatos = '-/Dropbox/TESIS/CODIGOS/COMPARACION ...
        MATRICIAL/CONJUNTO DE DATOS/Matrices_Creadas'; % ...
        direccion donde se van a guardar
13
14    for i = a % posiciones en filas
15        for j = a % posiciones en columnas
16            M = ventana(i, j, window, matriz,W); % creacion de ...
                la matriz con la ventana desplazada segun ...
                indique sus parametros
17
18            DirSalida = ...
                strcat(dirDatos, strcat('/matriz', num2str(cont)));
19            save(DirSalida, 'M'); % guardar la matriz creada en ...
                el directorio escogido
20            subplot(3,3,cont); % crea un plot con las matrices ...
                que se van creando conforme avanzan los indices.
21            imagesc(M);
22            %colorbar;
23            cont = cont + 1;
24        end
25    end
26    matrices = 0; % cero indica que se han creado satisfactoriamente
27 end

```

```
28
29 %funcion que coloca la ventana en su posicion indicada.
30 %posx = posicion en filas de la ventana
31 %posy = posicion en columnas de la ventana
32 %window = tamaño de la ventana
33 %matriz = matriz a la que se le creara la ventana
34 %W = valor del contenido de la ventana
35 function M = ventana(posx, posy, window, matriz,W)
36     M = matriz;
37     M(posx:posx + (window - 1), posy:posy + (window - 1)) = W;
38 end
```

Bibliografía

- [Angeles-Yreta, 2006] Angeles-Yreta, M.-A. (2006). Cómputo de la Similitud entre Figuras Geométricas. Tesis de máster, Centro de Investigación en Computación, Instituto Politécnico Nacional, México. (Citado en páginas 38, 40 y 42.)
- [Ashburner *et al.*, 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., Sherlock, G., y Consortium, G. O. (2000). Gene Ontology: tool for the unification of biology. *Nat. Genet.*, 25(1):25–29. (Citado en páginas 21 y 23.)
- [Atkinson y Shiffrin, 1968] Atkinson, R. C. y Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. *Psychol. Learn. Motiv. Adv. Res. theory*, 2:89–195. (Citado en página 10.)
- [Berners-Lee *et al.*, 2001] Berners-Lee, T., Hendler, J., y Lassila, O. (2001). The Semantic Web. *Sci. Am.*, 284(5):34–43. (Citado en página 22.)
- [Blei *et al.*, 2003] Blei, D. M., Ng, A., y Jordan, M. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022. (Citado en página 36.)
- [C. Bustillo-Hernández, 2009] C. Bustillo-Hernández, J. F.-N. (2009). Procedimiento para la Extracción y Normalización de Conocimiento en Humanos : Una

- Aproximación para la Anotación de Ontologías en la Semantic Web. En *ROCC*, pp. 1–6, Acapulco Guerrero, Mexico. IEEE seccion mexico. (Citado en página 27.)
- [Cabalín-Silva y Navarro-Hernández, 2008] Cabalín-Silva, D. y Navarro-Hernández, N. (2008). Conceptualización de los Estudiantes sobre el Buen Profesor Universitario en las Carreras de la Salud de la Universidad de La Frontera - Chile. *Int. J. Morphol.*, 26(4):887–892. (Citado en páginas 28 y 68.)
- [Chen y Chu, 2005] Chen, C. C. y Chu, H. T. (2005). Similarity measurement between images. En *Int. Comput. Softw. Appl. Conf.*, volumen 2, pp. 41–42. (Citado en página 56.)
- [Chouakria y Nagabhushan, 2007] Chouakria, A. D. y Nagabhushan, P. N. (2007). Adaptive dissimilarity index for measuring time series proximity. *Adv. Data Anal. Classif.*, 1(1):5–21. (Citado en página 51.)
- [Chu *et al.*, 2001] Chu, W., Sang-Wook, K., y Sanghyun, P. (2001). An index-based approach for similarity search supporting time warping in large sequence databases. *Proc. 17th Int. Conf. Data Eng.*, pp. 607–614. (Citado en página 49.)
- [Collins y Quillian., 1969] Collins, A. y Quillian., R. (1969). Retrieval Time from Semantic Memory. *J. Verbal Learning Verbal Behav.*, 8:240–247. (Citado en páginas 17, 18 y 28.)
- [Collins y Loftus, 1975] Collins, M. y Loftus, E. (1975). A spreading-Activation Theory of Semantic Processing. *Psychol. Rev.*, 82(6):407–428. (Citado en páginas 18, 28 y 36.)
- [Cristani y Verona, 2005] Cristani, M. y Verona, U. (2005). A Survey on Ontology Creation Methodologies. *Int. J. Semant. Web Inf. Syst.*, 1(2):49–69. (Citado en página 21.)

- [Deza y Deza, 2009] Deza, M. M. y Deza, E. (2009). *Encyclopedia of Distances*, volumen 3. Springer, 3 edición. (Citado en página 46.)
- [Dravnieks, 1982] Dravnieks, A. (1982). Odor quality: semantically generated multidimensional profiles are stable. *Science (80-.)*, 218(4574):799–801. (Citado en páginas 21 y 30.)
- [etorredebabel, 2016] etorredebabel (01 de Diciembre de 2016). <http://www.ettoredebabel.com/Historiadela-filosofia/Filosofiacontemporanea/Wittgenstein/Wittgenstein-JuegosLenguaje.htm>. (Citado en página 12.)
- [Figuroa, 1981] Figuroa, J. (1981). *Estudio de las Redes Semánticas Naturales y algunos de sus procesos básicos*. Facultad de Psicología de la UNAM. (Citado en página 72.)
- [Figuroa *et al.*, 2008] Figuroa, J., Angeles-Yreta, A., Medina-Apodaca, J., Ortega-González, V., Ramírez-Amaro, K., Mirón-Bernal, M., y Landassuri-Moreno, V. (2008). Sobre el problema de Semejanza. Technical report, Centro de Investigación en Computación. Instituto Politécnico Nacional Unidad Profesional “Adolfo López Mateos”, Zacatenco, México, DF. (Citado en página 39.)
- [Figuroa y Carrasco, 1980] Figuroa, J. y Carrasco, M. (1980). *Comparación de Red Semántica entre estudiantes ingleses y mexicanos*. Facultad de Psicología de la UNAM. (Citado en páginas 28 y 68.)
- [Figuroa *et al.*, 1981] Figuroa, J., Gonzalez, E., y Solis, V. (1981). Una aproximación al problema de significado: las Redes Semánticas. *Revista Latinoamericana de Psicología*, 13(3):447–458. (Citado en página 69.)

- [Figuerola *et al.*, 1976] Figuerola, J. G., González, E. G., y Solís, V. M. (1976). An Approach to the Problem of Meaning: Semantic Networks. *Journal of Psycholinguistic Research*, 5(2):107–115. (Citado en páginas 28, 32 y 68.)
- [Fréchet, 1906] Fréchet, M. M. (1906). Sur quelques points du calcul fonctionnel. *Rend. del Circ. Mat. di Palermo*, 22(1):1–72. (Citado en página 49.)
- [García-Cabrero y Jiménez-Vidal, 1996] García-Cabrero, B. y Jiménez-Vidal, S. (1996). Redes semánticas de los conceptos de presión y flotación en estudiantes de bachillerato. *Revista Mexicana de Investigación Educativa*, 1(2):343–361. (Citado en páginas 28 y 68.)
- [Genesereth y Nilsson, 1987] Genesereth, M. R. y Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann Publishers. (Citado en página 22.)
- [Goshtasby, 2012] Goshtasby, A. A. (2012). *Image Registration. Principles, Tools and Methods*. Springer-Verlag London, 1 edición. (Citado en página 47.)
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220. (Citado en página 21.)
- [Guarino, 1998] Guarino, N. (1998). Formal Ontology and Information Systems. En *Form. Ontol. Inf. Syst. Proc. FOIS'98.*, volumen 46, pp. 3–15, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press. (Citado en página 26.)
- [Jakus *et al.*, 2013] Jakus, G., Milutinović, V., Omerović, S., y Tomažič, S. (2013). *Concepts, Ontologies and Knowledge Representation*. Springer-Verlag New York, 1 edición. (Citado en página 22.)
- [Jepsen, 2009] Jepsen, T. C. (2009). Just what is an ontology, anyway? *IT Professional*, 11(5):22–27. (Citado en página 22.)

- [Keysers y Unger, 2003] Keysers, D. y Unger, W. (2003). Elastic image matching is NP-complete. *Pattern Recognit. Lett.*, 24(1-3):445–453. (Citado en página 49.)
- [Krantz *et al.*, 1971] Krantz, D. H., Luce, R. D., Suppes, P., y Tversky, A. (1971). *Foundation of Measurement - Additive and Polynomial Representation*, volumen I. Academic Press, New York, USA. (Citado en páginas 33 y 41.)
- [Krantz *et al.*, 1989] Krantz, D. H., Luce, R. D., Suppes, P., y Tversky, A. (1989). *Foundations of Measurement - Geometrical, Threshold and Probabilistic Representations*, volumen 2. Academic Press, New York. (Citado en páginas 33 y 41.)
- [Lakoff, 1987] Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, volumen 64. University of Chicago Press. (Citado en página 13.)
- [Landauer y Dumais, 1997] Landauer, T. K. y Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation Knowledge. *Psychol. Rev.*, 104(2):211–240. (Citado en página 35.)
- [Landauer *et al.*, 1990] Landauer, T. K., Dumais, S. T., Deerwester, S., y Furnas, G. (1990). Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.*, 41(6):391. (Citado en página 35.)
- [Landauer *et al.*, 1987] Landauer, T. K., Furnas, G. W., Dumais, S. T., y Gomez, L. M. (1987). The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971. (Citado en página 35.)
- [Lei y Govindaraju, 2004] Lei, H. y Govindaraju, V. (2004). Direct Image Matching by Dynamic Warping. En *2004 Conf. Comput. Vis. Pattern Recognit. Work. IEEE*, número 1. (Citado en página 52.)

- [Lin, 1998] Lin, D. (1998). An Information-Theoretic Definition of Similarity. *Proc. ICML*, pp. 296–304. (Citado en página 42.)
- [Luce *et al.*, 2006] Luce, R. D., Suppes, P., Krantz, D. H., y Tversky, A. (2006). *Foundations of measurement: Representation, axiomatization, and invariance*, volumen 3. Academic Press, New York. (Citado en páginas 33 y 41.)
- [Mandujano-García *et al.*, 2016] Mandujano-García, A., Figueroa, J., y Calvo, H. (2016). Comparación Cuantitativa de Matrices de Datos. En *CORE 2016, CIC IPN*, Ciudad de México, México. (Citado en página 95.)
- [Meyer y Schvaneveldt, 1976] Meyer, D. E. y Schvaneveldt, R. W. (1976). Meaning, memory structure, and mental processes. *Science*, 192(4234):27–33. (Citado en páginas 18 y 28.)
- [Ortega-González, 2007] Ortega-González, E.-V. (2007). Una Técnica para el Análisis de Similitud entre Imágenes. Tesis de máster, Centro de Investigación en Computación, Instituto Politécnico Nacional, México. (Citado en páginas 39 y 52.)
- [Pérez-Corona *et al.*, 2011] Pérez-Corona, N., Hernández-Colín, D., Bustillo-Hernández, C., y Figueroa-Nazuno, J. (2011). Sistema de Construcción de ontologías mediante la Teoría y Técnicas de Espacio Semántico Natural. En *17th Int. Congr. Comput. Sci. Res.*, pp. 1–12. (Citado en página 31.)
- [Pérez-Corona *et al.*, 2012] Pérez-Corona, N., Hernández-Colín, D., Bustillo-Hernández, C., y Figueroa-Nazuno, J. (2012). Model of Natural Semantic Space for Ontologies’ Construction. *Int. J. Comb. Optim. Probl. Informatics*, 3(2):93–108. (Citado en página 31.)
- [Petra-micu *et al.*, 2012] Petra-micu, I., Cortés-morelos, J., Talayero-uriarte, J. A., y Fouilloux-morales, M. (2012). Representación semántica del concepto “medicina”

- en estudiantes de la Facultad de Medicina , UNAM . Un estudio longitudinal. *Investigación en Educación Medica*, 1(2):49–56. (Citado en páginas 28 y 68.)
- [Quillian, 1967] Quillian, M. (1967). Word Concepts: a theory and simulation of some basic semantic capabilities. *Comput. Behav. Sci.*, 12:410–430. (Citado en páginas 18, 28 y 36.)
- [Quillian, 1969] Quillian, M. (1969). The Teachable Language Comprehender: A Simulation program and theory of Language. *Commun. Assn. Comp. Mach.*, 12(8):459—476. (Citado en páginas 18, 28 y 29.)
- [Rabiner y Juang, 1993] Rabiner, L. y Juang, B. (1993). *Fundamentals of Speech Recognition*. Prentice Hall. (Citado en página 49.)
- [Real Academia Española, 2016] Real Academia Española (10 de Agosto de 2016). <http://dle.rae.es/?id=KYtLWbc>. (Citado en página 7.)
- [Richens, 1958] Richens, R. H. (1958). Interlingual Machine Translation. *The Computer Journal*, 1(3):144–147. (Citado en página 17.)
- [Rosch y Mervis, 1975] Rosch, E. y Mervis, C. B. (1975). Family Resemblances: Studies in the Internal Structure of Categories. *Cogn. Psychol.*, 7:573–605. (Citado en página 14.)
- [Salvador y Chan, 2007] Salvador, S. y Chan, P. (2007). FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intell. Data Anal.*, 11:561–580. (Citado en página 51.)
- [Santos-Camacho, 2014] Santos-Camacho, E.-A. (2014). Ensamble y diseño de algoritmos para la clasificación no supervisada y de forma libre de series de tiempo. Tesis de máster, Centro de Investigación en Computación, Instituto Politécnico Nacional, México. (Citado en páginas XII, 44 y 48.)

- [Simmons, 1965] Simmons, R. (1965). Answering English questions by computer: a survey. *Commun. ACM*, 8(1):53–70. (Citado en página 17.)
- [Simmons, 1972] Simmons, R. (1972). Semantic Networks: their computation and use for understanding english sentences. Technical report, Technical Report NL-6, in Natural Language Research for Computer Assisted Instruction and Department of Computer Science, University of Texas. Supported by National Science Foundation. (Citado en página 17.)
- [Simmons, 1966] Simmons, R. F. (1966). Storage and retrieval of aspects of meaning in directed graph structures. *Commun. ACM*, 9(3):211–215. (Citado en página 17.)
- [Skinner, 1964] Skinner, B. F. (1964). New methods and new aims in teaching. *New Sci.*, 122:483–84. (Citado en página 9.)
- [Sowa, 2000] Sowa, J. F. (2000). *Knowledge Representation Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co. (Citado en página 17.)
- [Stevens, 1946] Stevens, S. (1946). Sobre la Teoría de las Escalas de Medición S. 103(2684):677–680. (Citado en página 33.)
- [Teknomo, Kardi, 2016] Teknomo, Kardi (12 de Agosto de 2016). Similarity measurement. <http://people.revoledu.com/kardi/tutorial/Similarity/index.html>. (Citado en páginas 39 y 43.)
- [Tversky, 1977] Tversky, A. (1977). Features of Similarity. *Psychol. Rev.*, 84:327–352. (Citado en página 42.)
- [Uchida y Sakoe, 1998] Uchida, S. y Sakoe, H. (1998). A monotonic and continuous two-dimensional warping based on dynamic programming. En *Proceedings. Fourteenth Int. Conf. Pattern Recognit. (Cat. No.98EX170)*, volumen 1. (Citado en página 54.)

- [Uchida y Sakoe, 2000] Uchida, S. y Sakoe, H. (2000). An Approximation Algorithm for Two-Dimensional. *IEICE TRANS. INF. SYST.*, E83-D(1):109–111. (Citado en página 48.)
- [Uchida y Sakoe, 2005] Uchida, S. y Sakoe, H. (2005). A survey of elastic matching techniques for handwritten character recognition. *IEICE Trans. Inf. Syst.*, E88-D(8):1781–1790. (Citado en páginas XII, 48, 53 y 54.)
- [Uschold y Jasper, 1999] Uschold, M. y Jasper, R. (1999). A Framework for Understanding and Classifying Ontology Applications. En *Proc. IJCAI-99 Work. Ontol. Probl. Stock. Sweden*, volumen 18, pp. 1–12, Stockholm, Sweden. (Citado en página 22.)
- [Vaiopoulos, 2011] Vaiopoulos, a. D. (2011). Developing Matlab scripts for image analysis and quality assessment. *Proc. SPIE*, 8181(2):81810B–81810B–12. (Citado en página 57.)
- [Valdez-Medina, 2004] Valdez-Medina, J. L. (2004). *Las Redes Semánticas Naturales. Usos y aplicaciones en Psicología Social*. Trillas, cuarta edición. (Citado en páginas 69 y 71.)
- [Vargas-Medina y Figueroa, 2007] Vargas-Medina, E. y Figueroa, J. (2007). Redes Semánticas Naturales y Construcción de Ontologías. En *Congreso Internacional, Metodología para la Educación a Distancia*, pp. 201–210. (Citado en páginas 10, 29 y 30.)
- [Wikipedia, 2016b] Wikipedia (01 de Diciembre de 2016b). https://es.wikipedia.org/wiki/Ludwig_Wittgenstein. (Citado en página 12.)
- [Wikipedia, 2016a] Wikipedia (12 de Noviembre de 2016a). Elicitación. <https://es.wikipedia.org/wiki/Elicitaci%C3%B3n>. (Citado en página 16.)

[Wittgenstein, 1922] Wittgenstein, L. (1922). *Tractatus logico-philosophicus*. (Citado en páginas 12 y 17.)

[Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical Investigations*. (Citado en página 12.)

[Zezula *et al.*, 2006] Zezula, P., Amato, G., Dohnal, V., y Batko, M. (2006). *Similarity Search, The Metric Space Approach*. Springer New York. (Citado en página 44.)