



upicisa

INSTITUTO POLITECNICO NACIONAL

**UNIDAD PROFESIONAL INTERDISCIPLINARIA
DE INGENIERIA Y CIENCIAS SOCIALES Y
ADMINISTRATIVAS**

**SISTEMA EVOLUTIVO PARA REPRESENTACION
DEL CONOCIMIENTO**

SEMINARIO DE TITULACION

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS
DE LA INFORMATICA
PRESENTA :
JESUS MANUEL OLIVARES CEJA**

Para este trabajo se eligió una Red Semántica Ampliada como formalismo para representar el conocimiento por ser una estructura general en contraparte con otras que son especializadas como los marcos, los guiones, etc (11).

II.1 RED SEMANTICA AMPLIADA (RSA).

Una Red Semantica Ampliada (RSA) [1] [9] [30] es un formalismo que enfatiza las relaciones entre objetos. Se compone por nodos que representan objetos y arcos dirigidos y etiquetados que representan relaciones. La semántica de cada nodo está dada por las relaciones que tiene con otros. Lleva el adjetivo "ampliada" porque el tipo de objetos que se representan son una extensión de los que se manejan en las redes semánticas tradicionales (etiquetas); aquí pueden ser además, conceptos, imágenes, rutinas semánticas, señales, etc. (figura 3).

Algunas de las formas que puede tener la RSA representan en forma natural jerarquías de objetos [17] [21], por ejemplo cuando algunos objetos tienen relaciones de subordinación con respecto a otro (figura 4a).

Otro ejemplo es la representación de supraordinación, cuando objetos de menor nivel integran (o forman parte de) un objeto de nivel superior (figura 4b).

(11) Por ejemplo, la Dependencia Conceptual (CD), es una estructura que representa relaciones entre los componentes de una acción; los Marcos, es una estructura que se usa para representar objetos complejos desde diferentes puntos de vista; los Guiones, representan secuencias de eventos comunes; los Procesos o Rutinas Semánticas, son útiles para representar el conocimiento de "cómo hacer" [16] [17] [30].



INSTITUTO POLITECNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERIA
Y CIENCIAS SOCIALES Y ADMINISTRATIVAS

THE NUM. 950
Tel. 5-79-94-22

IZTACALCO

MEXICO 8, D. F.

Dependencia JEF.LIC.CSAS.INF.

No. de Oficio ST.JLCI.327/91

Expediente _____

Enero 25, 1991.

ASUNTO: Autorización del Tema
de Titulación

OPCION: Seminario de Titulación

C. PASANTE
JESUS MANUEL OLIVARES CEJA
P r e s e n t e .

Tengo el agrado de comunicar a usted, el tema y contenido que se le ha autorizado desarrollar para la Opción de Seminario de Titulación: **"SISTEMA EVOLUTIVO PARA REPRESENTACION DEL CONOCIMIENTO"**.

INTRODUCCION	
CAPITULO I	SISTEMAS EVOLUTIVOS
CAPITULO II	REPRESENTACION DE CONOCIMIENTO
CAPITULO III	DESCRIPCION DEL PROTOTIPO
CONCLUSIONES	
ANEXOS	
BIBLIOGRAFIA	

El trabajo de Titulación le será dirigido por el LIC. FERNANDO GALINDO SORIA.

A T E N T A M E N T E
"LA TECNICA AL SERVICIO DE LA PATRIA"

LIC. SUSANA JAUREGUI HONORATO
JEFE DE LA LICENCIATURA EN CIENCIAS
DE LA INFORMATICA
SUBDIRECCION DE LA
JEFATURA DE CARRERA
INFORMATICA

SJH'gmm*

DEDICATORIA:

A la conciencia de las personas.

AGRADECIMIENTOS:

Lo que hay atrás de un logro los
participantes lo conocen.

INDICE

	<u>Pág.</u>
INTRODUCCION	1
CAPITULO I.	
SISTEMAS EVOLUTIVOS	3
I.1 Definición	4
I.2 Características	5
I.3 Elementos	5
I.4 Ejemplos de Sistemas Evolutivos	12
I.5 Sistema Evolutivo para Representación del Conocimiento	12
CAPITULO II.	
REPRESENTACION DE CONOCIMIENTO	28
II.1 Red Semántica Ampliada (RSA)	29
II.2 Abstracción	31
II.3 Crecimiento de la RSA por agregación de objetos	33
II.4 Aplicación de la Inferencia Gramatical	43
CAPITULO III.	
DESCRIPCION DEL PROTOTIPO	49
III.1 Constructor Léxico	51
III.2 Constructor Sintáctico	56
III.3 Manejador de Gramática con Atributos	61

III.4 Generalizador	64
-------------------------------	----

III.5 Administrador	68
-------------------------------	----

CONCLUSIONES	71
------------------------	----

ANEXOS	72
------------------	----

1: Gramáticas	72
-------------------------	----

2: Gramática con Atributos del Prototipo	76
--	----

3: Modelo de agregación	78
-----------------------------------	----

4: Fractal de un árbol	86
----------------------------------	----

BIBLIOGRAFIA	88
------------------------	----

En el primer capítulo, se anuncia lo que se entiende por Sistema Evolutivo, sus características, elementos, ejemplos de ellos y se ilustra el de este informe.

- INDICACIONES.- Que se hace por inducción [1].
 INDICACION.- Modo de razonar que consiste en sacar de los hechos particulares una conclusión general [1].
 DEDUCTIVO.- Que obra por deducción [1].
 DEDUCTIVO.- Acción de deducir [1].
 DEDUCTIVO.- Sacar consecuencias. Conclusión [1].

INTRODUCCION

Este documento contiene los conceptos, propuesta y desarrollo de un Sistema Evolutivo cuyo fundamento principal son las investigaciones que sobre este tema se realizan en la UPIICSA. En estos sistemas se enfatiza el enfoque inductivo (1), en el que un mecanismo, con base a un número suficiente de ejemplos busca patrones que se repiten y propone las reglas o secuencia de actividades que posteriormente se pueden usar en su complemento deductivo (2), que aplica las reglas o pasos para obtener soluciones de problemas particulares que se le indiquen.

El título de este trabajo se considera en el sentido de representar de alguna forma conocimiento en término de relaciones entre objetos, conceptualizados y representados formalmente con herramientas matemáticas como la distribución, factorización y el concepto de gramática. Lo cual se busca simular en una Red Semántica a nivel de manejo de objetos.

En el primer capítulo, se enuncia lo que se entiende por Sistema Evolutivo, sus características, elementos, ejemplos de ellos y se ilustra el de este informe.

(1) INDUCTIVO.- Que se hace por inducción [11].

INDUCCION.- Modo de razonar que consiste en sacar de los hechos particulares una conclusión general [11].

(2) DEDUCTIVO.- Que obra por deducción [11].

DEDUCCION.- Acción de deducir [11].

DEDUCIR.- Sacar consecuencias. Conclusión [11].

En el segundo capítulo se comenta la representación de conocimiento en una RSA, en la que se usa la noción de fractal para caracterizar las formas obtenidas al agregarse objetos. La factorización se emplea para obtener hechos que se repiten y pueden proponerse como "reglas" o generalizaciones.

En el tercer capítulo se esquematiza la estructura general del prototipo y se explican algunos de sus módulos principales.

Las conclusiones se refieren a aplicaciones y extensiones del prototipo.

El anexo 1, es una breve introducción a las gramáticas. El anexo 2, muestra la gramática con atributos completa que se emplea en el prototipo. En los anexos 3 y 4 se muestran los listados de los programas de un modelo de agregación y de un fractal de árbol.

Las referencias bibliográficas que en el texto se señalan entre corchetes aparecen al final.

CAPITULO I. SISTEMAS EVOLUTIVOS

"...el universo se compone de materia,
energía e información formando una unidad..."

Fernando Galindo Soria

El antecedente de los Sistemas Evolutivos es la Programación Dirigida por Sintaxis (PDS) [13]; la cual es una metodología que permite construir sistemas mediante la inducción de reglas generales a partir de ejemplos específicos, para aplicarlas a la solución de problemas particulares [5]. La base de este método es el estudio del lenguaje del área de aplicación, considerando que al observar el lenguaje que se utiliza en un área se encuentran en forma natural los componentes [3] de su sistema de información.

La PDS se apoya en la equivalencia estructural entre gramáticas y sistemas de información que se resumen de [13] como sigue:

PROGRAMA TIPO	ESTRUCTURAS DE PROGRAMACION	GRAMATICA
Ejecuta B Ejecuta C	Secuencia	$A \rightarrow B C$
Si C entonces D sino E	Decisión	$A \rightarrow C D \mid E$
Mientras F Ejecuta G Ejecuta Y	Iteración	$A \rightarrow X$ $X \rightarrow F G X \mid Y$
Si Z entonces Llama a A Ejecuta Y	Recursión	$A \rightarrow Z A \mid Y$

Las fases [13] de este método son:

a) análisis.- comprende las etapas de:

- 1) Identificación del área de aplicación
- 2) Estudio del lenguaje y detección de ejemplos significativos (considerando la problemática a resolver)

b) diseño.- con:

- 1) Detección de unidades léxicas (3) del lenguaje utilizado en el área de aplicación
- 2) Obtención de una gramática generativa (4) mediante la aplicación de la Inferencia Gramatical sobre las oraciones canónicas (5) de los ejemplos del lenguaje obtenidos en el análisis
- 3) Depuración y obtención de una gramática con atributos

c) implantación.- Se utiliza una componente deductiva como: un Sistema Experto, un generador de sistemas, una aplicación en Prolog, etc, o bien, se obtiene un programa directamente utilizando las equivalencias estructurales entre gramáticas y sistemas de información mencionadas arriba.

I.1 DEFINICION.

Un Sistema Evolutivo [3] [4] es una herramienta automatizada "capaz de construir y mantener en tiempo real una imagen del ambiente que le rodea" mediante la adquisición de conocimiento, "como si fuera un ser vivo que va adquiriendo conocimiento" mediante la interacción con su entorno. Esta herramienta debe ser fácil de construir y de utilizar.

(3) Son las unidades mínimas con significado en el lenguaje. Se clasifican en tipos el cual es un símbolo que identifica a un grupo de ellas, por ejemplo, acciones, objetos, preposiciones, etc. [13]

(4) Es una gramática con producciones para generar un lenguaje determinado.

I.2 CARACTERISTICAS.

Un Sistema Evolutivo se caracteriza [2] por sus capacidades de:

- a) captación de la realidad
- b) almacenamiento y representación de conocimiento
- c) abstracción (6) e inferencia de reglas a partir de un conjunto de conocimientos
- d) mantenimiento en tiempo real de la imagen de la realidad
- e) diálogo con el exterior.

Por sus características un Sistema Evolutivo resuelve algunas de las mayores dificultades actuales de los sistemas de información en lo referente al modelado de la realidad:

- a) representación de la realidad [3] [7]
- b) mantenimiento automático del modelo [3] [7]
- c) separación explícita de los componentes de un sistema de información (Independencia Relativa) [4]
- d) oportunidad en la obtención y actualización del modelo [4].

I.3 ELEMENTOS.

En general se distinguen dos grupos de elementos en un Sistema Evolutivo [6] (Figura 1):

- a) estáticos
- b) dinámicos.

(5) Es la que resulta de sustituir cada unidad léxica de una oración por su tipo correspondiente.

(6) La definición de este término se da en la sección II.2.

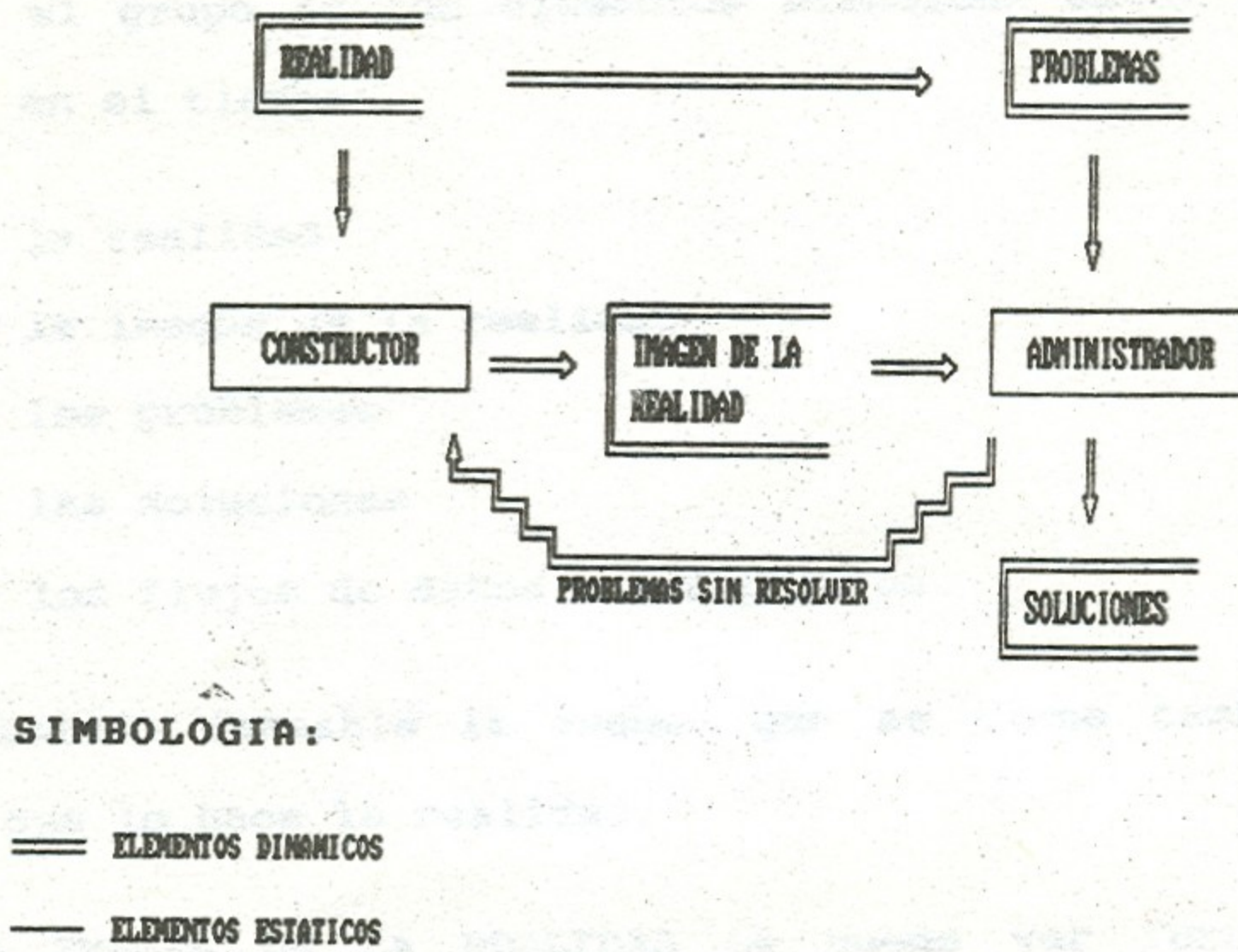


Figura 1.- Elementos de un Sistema Evolutivo

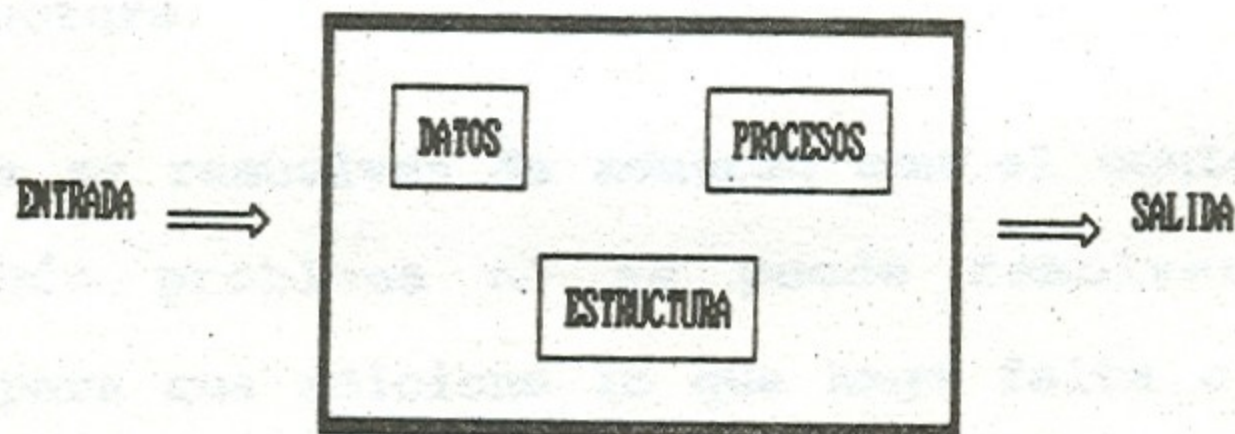


Figura 2.- Componentes generales de un Sistema de Información

En el grupo de los elementos **DINAMICOS** están aquellos que cambian en el tiempo:

- a) la realidad
- b) la imagen de la realidad
- c) los problemas
- d) las soluciones
- e) los flujos de datos e información.

En un estado deseable la imagen que se tiene cambia al mismo tiempo que lo hace la realidad.

La **IMAGEN DE LA REALIDAD** se puede ver formada por los componentes generales de un sistema de información, mediante los cuales se puede modelar ésta [4] (Figura 2):

- a) datos
- b) procesos
- c) estructura.

Los problemas se resuelven de acuerdo con el contenido de ésta; pero si algún problema no se puede resolver se pasa al constructor para que adicione lo que haga falta o determine que es un problema que no pertenece a su ámbito.

El manejo de los datos puede estar a cargo de una base de datos o de conocimientos. Los procesos se pueden manejar mediante macros, funciones o algún generador de sistemas. La estructura del sistema puede describirse mediante una gramática con atributos.

Los elementos **ESTATICOS** aportan las características de evolución, aprendizaje y coordinación de sus similares dinámicos. Estos permanecen igual conforme el sistema opera, son:

- a) el constructor
- b) el administrador.

El **CONSTRUCTOR** es el elemento inductivo que obtiene la imagen de la realidad, mediante la captación y el reconocimiento de patrones [2]. Para esto se auxilia de la abstracción, y de operaciones de Inferencia Gramatical. Algunas implementaciones tienen:

- a) un constructor léxico
- b) un constructor sintáctico
- c) un constructor semántico.

El **CONSTRUCTOR LEXICO** [4] recibe como entrada una oración en algún lenguaje, obtiene las unidades léxicas presentes, el tipo de cada una y la estructura u oración canónica.

Ejemplo:

Dada la oración "CALCULA LA REGRESION DE X * Y + Z" se obtiene

UNIDAD LEXICA

TIPO

CALCULA
LA
REGRESION
DE
X
*
Y
+
Z

Acción (a)
Ignorable (i)
Acción (a)
Ignorable (i)
Dato (d)
Acción (a)
Dato (d)
Acción (a)
Dato (d)

junto con la oración canónica "a i a i d a d a d".

Este componente inicia su operación sin conocer nada acerca del lenguaje a utilizar, únicamente cuenta con la características mencionadas.

Algunas estrategias [4] utilizadas para su implementación son:

- a) Unidades predefinidas.- Se le da al inicio una tabla que va incrementando al obtener datos nuevos
- b) Carácteres especiales.- Se usan carácteres especiales que permiten identificar el tipo
- c) Manejo de contexto.- Se usa el contexto para determinar el tipo
- d) Mecanismo de diálogo.- Encuentra en tiempo real las unidades léxicas nuevas y les asigna su significado
- e) Otros mecanismos mas poderosos requieren del apoyo sus análogos sintáctico y semántico.

El CONSTRUCTOR SINTACTICO recibe las oraciones canónicas obtenidas y si se desconocen las integra a su gramática generativa.

Cuando inicia no cuenta con una gramática definida, así que al conjunto de canónicas les aplica alguna o varias operaciones de Inferencia Gramatical (7) que consisten básicamente en factorización, recursividad y distribución (8).

En la factorización se busca un elemento común entre los casos que se tienen:

(7) Es una herramienta de la Lingüística Matemática que mediante operaciones sobre un conjunto de oraciones significativas de un lenguaje, permite encontrar su gramática generativa. Se inició utilizando en el área de Reconocimiento de Patrones y se ha extendido su aplicación en la construcción de Sistemas Evolutivos [14].

(8) Esta operación se describe en III.2 en este documento.

Ejemplo:

Sea la gramática canónica (9)

$$S \rightarrow a b c d \mid a b m e \mid a b j f$$

factorizando "a b" en la gramática canónica se obtiene

$$\begin{aligned} S &\rightarrow a b X \\ X &\rightarrow c d \mid m e \mid j f \end{aligned}$$

Algunos casos [14] de esta operación son:

- a) izquierda
- b) derecha
- c) empotrada
- d) multiempotrada.

En la recursividad [14] se tiene un **Factor de repetición** y un **Punto de convergencia** donde termina ésta.

Ejemplo:

Sea la gramática canónica

$$S \rightarrow a b \mid a a b \mid a a a b$$

de la que se obtiene el factor de repetición "a" y el punto de convergencia "b" y por lo tanto su gramática generativa es:

$$S \rightarrow a S \mid b$$

Dado que esta gramática es mas general que la original, en su momento puede llegar a generar cadenas inválidas.

Algunos casos [14] de esta operación son:

- a) izquierda
- b) derecha
- c) empotrada
- d) multiempotrada
- e) diferida.

(9) Es el conjunto de las oraciones canónicas de los ejemplos significativos del lenguaje [15].

I.4 EJEMPLOS DE SISTEMAS EVOLUTIVOS.

Algunos Sistemas Evolutivos propuestos en la UPIICSA son:

- a) Lenguajes de Trayectoria [1].-considera a la trayectoria entre dos puntos cualesquiera como una oración
- b) Generador de Esquemas Lógicos de Base de Datos [5] [25].- obtiene un esquema de base de datos a partir de la descripción con oraciones en lenguaje natural de un ambiente
- c) EVA [6].- es un concepto que plantea "sumar" características evolutivas a los sistemas existentes, con el proposito de darles la capacidad de evolucionar
- d) Constructor de Sistemas Expertos [10].- explica cómo construir Sistemas Expertos utilizando Inferencia Gramatical
- e) Manejador de Paisajes (realizado por Carlos Olicón Nava).- crea y maneja paisajes mediante oraciones en lenguaje natural

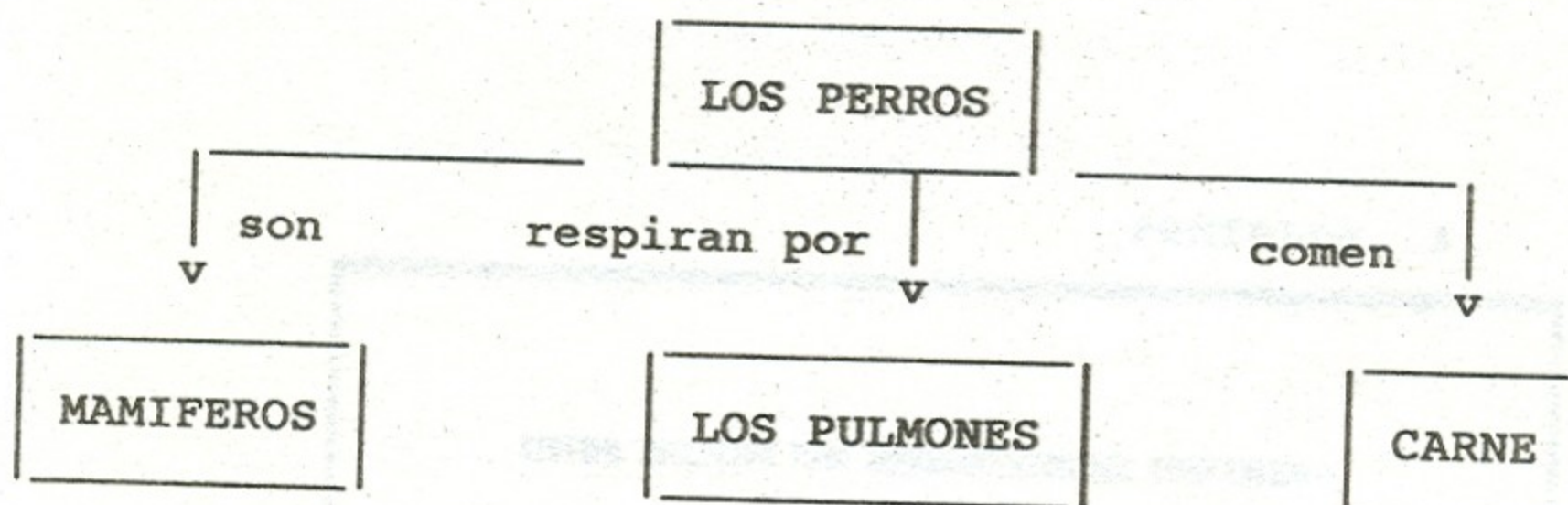
I.5 SISTEMA EVOLUTIVO PARA REPRESENTACION DEL CONOCIMIENTO

Este sistema a partir de oraciones declarativas en lenguaje natural obtiene una Red Semántica Ampliada apoyandose en la Inferencia Gramatical. Los hechos de las oraciones que se le dan los integra directamente en la red y accedandola se puede solicitar la busqueda de hechos que se repiten un número suficiente de veces para proponer un objeto generico. El Administrador se ejemplifica con la opción de consulta de información registrada, mediante oraciones sencillas de tipo interrogativo e imperativo.

Un ejemplo de oración declarativa que se puede dar como entrada es:

LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE

de esta oración se obtiene:



de la que una posible consulta puede ser:

Usuario: ¿QUE SON LOS PERROS?
 Sistema: SON MAMIFEROS

A continuación se ilustra la forma en que opera usando las pantallas con las que interactúa el usuario.

Cada pantalla lleva una breve explicación en la parte superior.

Cuando se referencia una tecla se indica su nombre con letras mayúsculas entre "<" y ">".

Siempre que se de un dato se debe oprimir la tecla <ENTER> para que se acepte.

Para salir de las ventanas sin seleccionar algún dato se usa la tecla <ESCAPE>.

Las pantallas que muestran algo de interés llevan un asterisco adelante de su número (por ejemplo 19*).

Si en la oracion hay partes desconocidas se puede indicar alguna unidad lexica con las teclas <F1> y <F2>(véase la pantalla 7).

Para continuar se oprime la tecla <F8>. Si se reconoce la oración el sistema sigue en la pantalla 4, de otra forma sigue en la 15.

Si se quiere cancelar la oración, se oprime la tecla <ESCAPE>.

PANTALLA 5*

Marque alguna unidad lexica	
F1-Inicio, F2-Fin, F3-Unidades Lexicas, F8-Continua	
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	

En la pantalla 5 se pueden ver las unidades léxicas que el sistema tiene registradas oprimiendo la tecla <F3>.

Para quitar la ventana se oprime la tecla <ESCAPE>.

PANTALLA 6

Marque alguna unidad léxica	
F1-Inicio, F2-Fin, F3-Unidades Lexicas, F8-Continua	
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
NUMERO	UNIDAD LEXICA
1	BIOLOGIA
2	LOS
3	OCELOTES
4	SON
5	ESTUDIA A
6	SE DIVIDEN EN

La unidad léxica seleccionada aparece abajo de la oración.
Se debe indicar el atributo de la unidad léxica, que puede ser su tipo (un carácter) o algún número de sinónimo.

PANTALLA 7*

Escriba su atributo(tipo o sinónimo)	
F4=Tipos, F5=Sinónimos	
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO

Si la unidad léxica que se seleccionó ya existe el sistema lo indica con un aviso.

Después de oprimir una tecla el sistema continua en la pantalla 5.

PANTALLA 8

AVISO: YA EXISTE ESTA UNIDAD LEXICA, oprima una tecla	
Escriba su atributo(tipo o sinónimo)	
F4=Tipos, F5=Sinónimos	
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO

En la pantalla 7 se indica el tipo de la unidad léxica con una letra o algún carácter especial.

Después de aceptar el tipo, el sistema sigue en la pantalla 5.

PANTALLA 9

Escriba su atributo(tipo o sinónimo)	
F4=Tipos, F5=Sinónimos	
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO r

Si el tipo es nuevo se pide su nombre; este dato es informativo.

Después de aceptar este dato el sistema pasa a la pantalla 5.

PANTALLA 10

Escriba el nombre del tipo		
F4=Tipos, F5=Sinónimos		
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE		
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO r	NOMBRE DEL TIPO -

En la pantalla 7 se puede indicar un número de sinónimo existente.
Después de aceptar el dato válido, el sistema pasa a la pantalla 5.

PANTALLA 11

Escriba su atributo(tipo o sinónimo)	
F4=Tipos, F5=Sinónimos	
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO 12

Si el sinónimo de la pantalla 11 es inválido se indica esto con un aviso.

Después de oprimir una tecla el sistema sigue en la pantalla 7.

PANTALLA 12

AVISO: SINONIMO FUERA DE RANGO, oprima una tecla	
Escriba su atributo(tipo o sinónimo)	
F4=Tipos, F5=Sinónimos	
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO 79432

En la pantalla 7 se pueden solicitar los tipos existentes con la tecla <F4>. En la ventana se puede seleccionar alguno con la tecla <ENTER> y el sistema continua en la pantalla 7; o se puede salir sin hacer seleccion oprimiendo la tecla <ESCAPE> con lo que se continua en la pantalla 7.

PANTALLA 13

Escriba su atributo(tipo o sinónimo)																
F4=Tipos, F5=Sinónimos																
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE																
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO	<table border="1"> <thead> <tr> <th>CLAVE</th> <th>NOMBRE</th> </tr> </thead> <tbody> <tr><td>i</td><td>Ignora</td></tr> <tr><td>o</td><td>Objeto</td></tr> <tr><td>r</td><td>Relacion</td></tr> <tr><td>f</td><td>Figura</td></tr> <tr><td>p</td><td>Pregunta</td></tr> <tr><td>q</td><td>Pregunta de quien</td></tr> </tbody> </table>	CLAVE	NOMBRE	i	Ignora	o	Objeto	r	Relacion	f	Figura	p	Pregunta	q	Pregunta de quien
CLAVE	NOMBRE															
i	Ignora															
o	Objeto															
r	Relacion															
f	Figura															
p	Pregunta															
q	Pregunta de quien															

En la pantalla 7 se pueden solicitar los sinónimos existentes con la tecla <F4>. En la ventana se puede seleccionar alguno con la tecla <ENTER> y el sistema sigue en la pantalla 5; o salir sin hacer selección oprimiendo la tecla <ESCAPE> con lo que se continua en la pantalla 7.

PANTALLA 14

Escriba su atributo(tipo o sinónimo)																								
F4=Tipos, F5=Sinónimos																								
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE																								
UNIDAD LEXICA RESPIRAN POR	ATRIBUTO	<table border="1"> <thead> <tr> <th>NUMERO</th> <th>SINONIMO</th> <th>TIPO</th> </tr> </thead> <tbody> <tr><td>9</td><td>TIENE</td><td>r</td></tr> <tr><td>10</td><td>ZANAHORIA</td><td>o</td></tr> <tr><td>11</td><td>LAS</td><td>i</td></tr> <tr><td>12</td><td>OSOS</td><td>o</td></tr> <tr><td>13</td><td>CON</td><td>i</td></tr> <tr><td>14</td><td>ESTA</td><td>r</td></tr> </tbody> </table>	NUMERO	SINONIMO	TIPO	9	TIENE	r	10	ZANAHORIA	o	11	LAS	i	12	OSOS	o	13	CON	i	14	ESTA	r	
NUMERO	SINONIMO	TIPO																						
9	TIENE	r																						
10	ZANAHORIA	o																						
11	LAS	i																						
12	OSOS	o																						
13	CON	i																						
14	ESTA	r																						

En la pantalla 15 se le indica al sistema que aplique la DISTRIBUCION en la oración oprimiendo la tecla (F3) con lo que el sistema pasa a la pantalla 15 (véase el capítulo III en este documento).

PANTALLA 19*

Indique la transformación a utilizar	
F1=Original, F2=Asignar sinónimo, F3=Distribución, F8=Continua	
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
ORACION CANONICA o1 r1 o2 r2 o3 r3 o4	
ORACION DISTRIBUIDA o1 r1 o2 ; o1 r2 o3 ; o1 r3 o4 .	

Si la oración canónica transformada no se reconoce por la gramática con atributos, el sistema envía un aviso indicándolo.

Después de oprimir una tecla el sistema sigue en la pantalla 3.

PANTALLA 20*

AVISO: ORACION IRRECONOCIBLE, oprima una tecla
LOS PERROS, LOS PULMONES Y COMEN CARNE

El generalizador se activa con una unidad léxica de tipo "g".

PANTALLA 21*

Escriba su oración	
ESC =Termina	
GENERALIZA	

Los hechos que se proponen como generales se indican opriniendo la tecla (F1), el sistema sigue en la pantalla 23.

Si no se quiere seleccionar alguno se oprime la tecla (ESCAPE) y se continua en la pantalla 4.

PANTALLA 22

Oprima 'F1' para seleccionar los hechos a generalizar	
GENERALIZA	
HECHOS COMUNES #1 SON Y RESPIRAN POR	MAMIFEROS PULMONES

Para seleccionar el objeto genérico aparece la ventana de unidades léxicas, se ubica con el cursor y luego se oprime la tecla <ENTER>. Si no existe se oprime la tecla <ESCAPE>, y el sistema pasa a la pantalla 4.

El usuario puede indicar la unidad léxica en la pantalla 3 y luego volver a activar el generalizador.

PANTALLA 23

Seleccione el objeto genérico			
GENERALIZA			
NUMERO	UNIDAD LEXICA	TIPO	SINON.
1	BIOLOGIA	o	1
2	LOS	i	2
3	OCELOTES	o	3
4	SON	r	4
5	ESTUDIA A	r	5
6	SE DIVIDEN EN	r	6

Para indicarle al sistema la relación con la que se unen los objetos particulares con el generico se realiza la operacion de manera análoga que en la pantalla 23.

Despues de aceptar la relación el sistema sigue en la pantalla 4.

PANTALLA 24

Seleccione la relación genérica			
GENERALIZA			
NUMERO	UNIDAD LEXICA	TIPO	SINON.
1	BIOLOGIA	o	1
2	LOS	i	2
3	OCELOTES	o	3
4	SON	r	4
5	ESTUDIA A	r	5
6	SE DIVIDEN EN	r	6

Las oraciones interrogativas e imperativas al interpretarse despliegan en la pantalla la respuesta al requerimiento.

Al terminar de indicar la respuesta el sistema sigue en la pantalla 4.

PANTALLA 25*

DIME QUIENES SON MANIFEROS LOS PERROS LOS GATOS LOS HOMBRES oprima una tecla

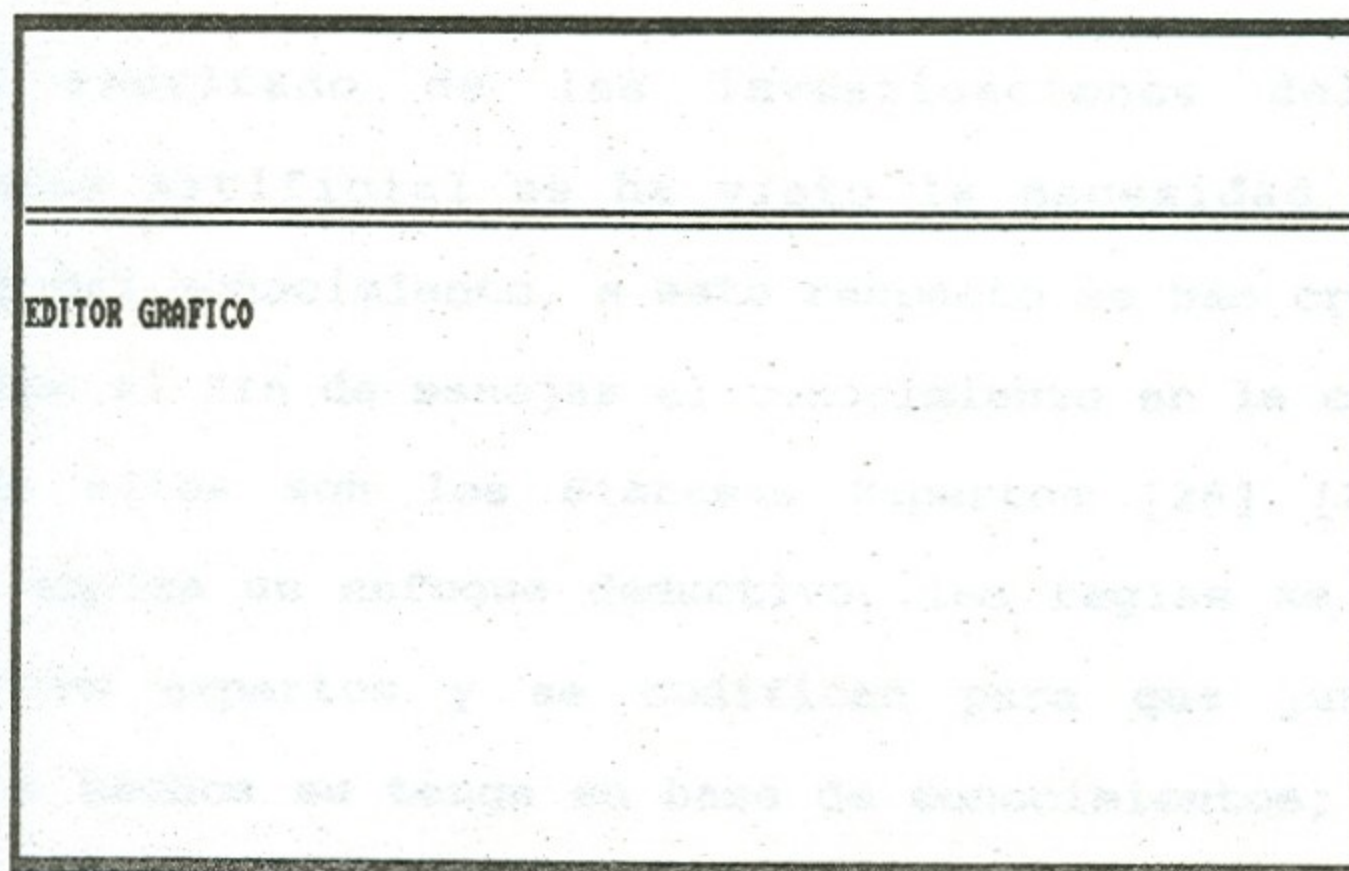
Para ver el contenido de la RSA se indica alguna unidad léxica de tipo "n".
Despues de mostrarla el sistema sigue en la pantalla 4.

PANTALLA 26*

MUESTRA LA RSA

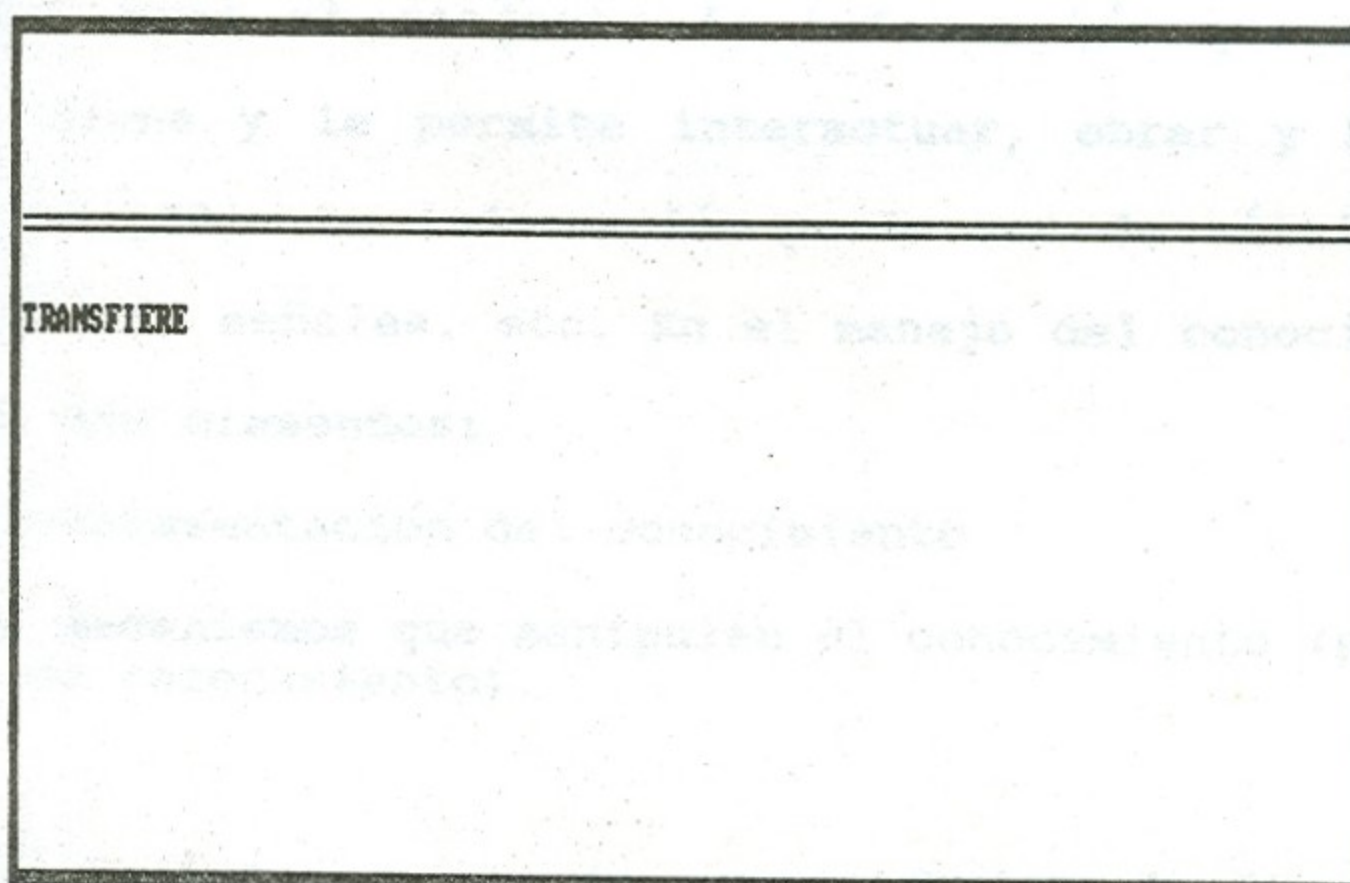
Para activar el editor gráfico se indica alguna unidad léxica de tipo "d".

PANTALLA 27



El convertidor de imágenes se activa con una unidad léxica de tipo "c".

PANTALLA 28



La pantalla inicial presenta al programa y al autor.

PANTALLA 1

SISTEMA EVOLUTIVO PARA REPRESENTACION DEL CONOCIMIENTO
Version 1.0 Enero 1991

Cualquier comentario o sugerencia se agradecera si se envía a:

JESUS MANUEL OLIVARES CEJA
Av. Rio Mayo 21-A
Fracc. Real del Moral
09010, Mexico, D. F., MEXICO

Para facilitar la referencia la pantalla se divide en áreas:

- A) avisos
- B) instrucciones
- C) opciones
- D) zona de trabajo

PANTALLA 2

AREAS

A

B

C

D

En esta pantalla se escribe una oración.

De las oraciones declarativas se toman los hechos para integrarlos a la RSA.

Las oraciones interrogativas e imperativas permiten consultar el contenido de la RSA.

Para salir del sistema se oprime la tecla <ESCAPE>.

PANTALLA 3*

Escriba su oracion	
ESC.-Termina	

Si la oración se reconoce y se interpreta correctamente el sistema lo indica con un aviso.

Despues de oprimir una tecla, el sistema continua en la pantalla 3.

PANTALLA 4*

AVISO: ORACION RECONOCIDA, oprima una tecla	
LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	

El programa obtiene la oración canónica y si es desconocida se solicita la transformación a aplicarle.

Después de indicar la transformación se debe oprimir <F8> con lo que el sistema pasa a la pantalla 4 o a la 28 según se reconozca o no la oración.

Oprimiendo <ESCAPE> se cancela la oración y el sistema pasa a la pantalla 3.

PANTALLA 15*

Indique la transformación a utilizar F1=Original, F2=Asignar sinónimo, F3=Distribución, F8=Continua LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE ORACION CANONICA o1 r1 o2 r2 o3 r3 o4
--

En la pantalla 15 se indica que se utilice la misma oración canónica obtenida oprimiendo la tecla <F1>.

PANTALLA 16

Indique la transformación a utilizar F1=Original, F2=Asignar sinónimo, F3=Distribución, F8=Continua LOS PERROS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE ORACION CANONICA o1 r1 o2 r2 o3 r3 o4 SE USA LA ORIGINAL
--

En la pantalla 15 se puede indicar una transformación determinada por el usuario, oprimiendo la tecla <F2>.

PANTALLA 17

Escriba el sinónimo de la estructura	
F5=Sinónimos	
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
ORACION CANONICA o1 r1 o2 r2 o3 r3 o4	
SINONIMO -	

En la pantalla 17 se pueden solicitar los sinónimos de estructuras existentes oprimiendo la tecla <F5>.

En la ventana se puede seleccionar alguno oprimiendo la tecla <ENTER> y el sistema sigue en la pantalla 15; o salir sin seleccionar nada oprimiendo la tecla <ESCAPE> con lo que se continua en la pantalla 17.

PANTALLA 18

Escriba el sinónimo de la estructura	
F5=Sinónimos	
LOS PERROS SON MANIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE	
ORACION CANONICA o1 r1 o2 r2 o3 r3 o4	
NUMERO	SINONIMO DE LA ESTRUCTURA
1 o2 r1 o1 .	
2 o1 r2 o3 ; o2 r1 o3 .	
3 o3 r1 o2 ; o3 r1 o1 .	

CAPITULO II. REPRESENTACION DE CONOCIMIENTO

El ser humano percibe el universo de manera abstracta

Jesús Manuel Olivares Ceja

Como resultado de las investigaciones del área de Inteligencia Artificial se ha visto la necesidad del manejo automático del conocimiento, a este respecto se han creado varios sistemas con el fin de manejar el conocimiento en la computadora, algunos de ellos son los Sistemas Expertos [26] [27] en los cuales se emplea un enfoque deductivo, las reglas se extraen de uno o varios expertos y se codifican para que junto con un conjunto de hechos se tenga su base de conocimientos; y entonces puede hacer deducciones mediante el motor de inferencia.

Otros sistemas que manejan el conocimiento son los Sistemas Evolutivos, en éstos se emplea la abstracción y la inferencia para la detección de reglas y generalizaciones. A partir de éstos puede construirse un Sistema Experto [10]. El conocimiento se maneja aquí como el conjunto de información que alguien (o algo) (10) tiene y le permite interactuar, obrar y hacer en su realidad [11] [30]. La información puede ser de múltiples formas: texto, imágenes, señales, etc. En el manejo del conocimiento son importantes dos elementos:

- a) la representación del conocimiento
- b) los mecanismos que manipulan el conocimiento (por ejemplo, mecanismos de razonamiento).

(10) Se dice "algo" en vista de que en el área de IA las máquinas pueden, aunque en forma rudimentaria, manejar el conocimiento.

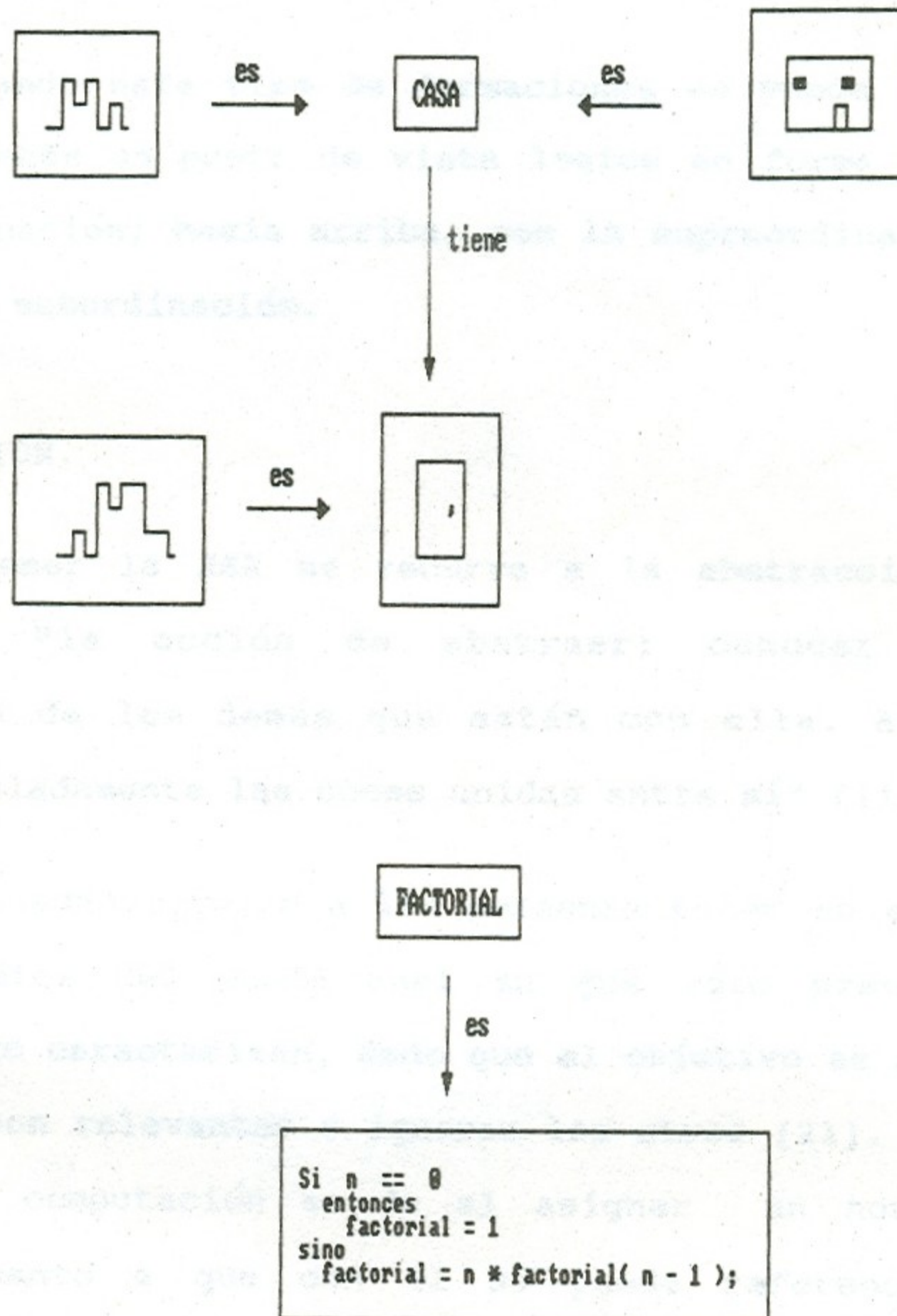


Figura 3.- Una Red Semántica Ampliada

Un tercer ejemplo es la coordinación, que se da entre elementos del mismo nivel, los que pueden no estar relacionados entre sí, como en la figura 4a (asiento, respaldo y café), o bien estarlo como en el caso en que dos o más objetos se explican en términos de otro equivalente (figura 4c).

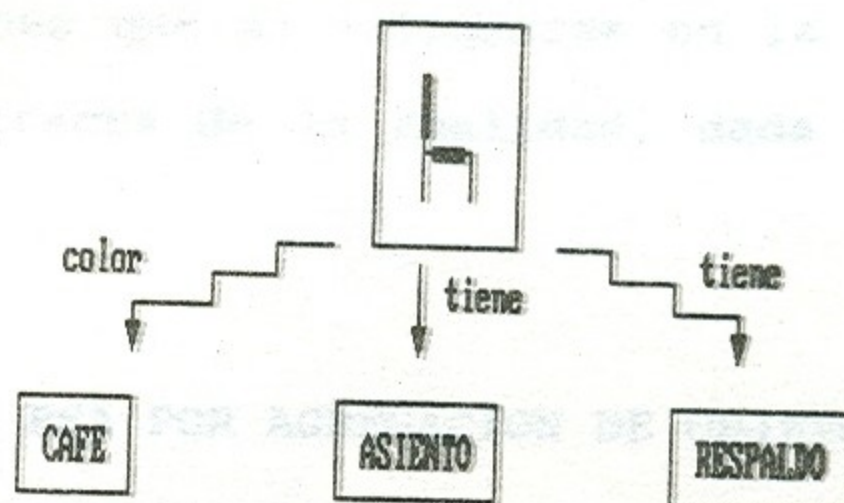
Considerando este tipo de formaciones se puede observar un crecimiento desde un punto de vista lógico en forma horizontal, con la coordinación; hacia arriba, con la supraordinación; hacia abajo, con la subordinación.

II.2 ABSTRACCION.

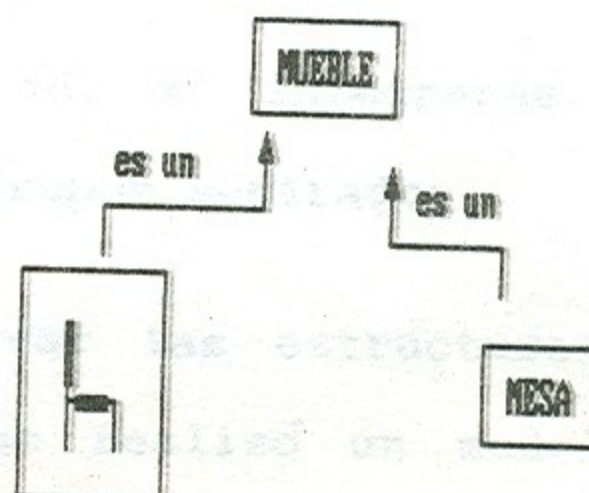
Para obtener la RSA se recurre a la abstracción, que se define como "la acción de abstraer; conocer una cosa prescindiendo de las demás que están con ella. Abstraer es considerar aisladamente las cosas unidas entre sí" [11].

Esta propiedad permite a las personas poner en el plano de la mente objetos del mundo real en que sólo prevalecen los detalles que lo caracterizan, dado que el objetivo es manejar los detalles que son relevantes e ignorar los otros [21]. Un ejemplo de su uso en computación se da al asignar un nombre a una rutina, en cuanto a que con él se puede referenciar a los elementos que la forman, esto evita tener que referirse a los datos, instrucciones y estructura de control en forma explícita. En general ésta se usa en computación para hacer modelos más manejables [22] [4].

a) Subordinación



b) Supraordinación



c) Coordinación

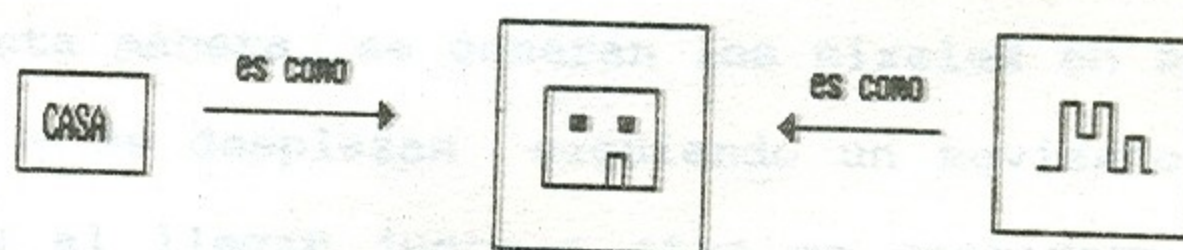


Figura 4.- Formas jerárquicas en la RSA

En este trabajo permite captar en forma independiente los objetos y las relaciones que al integrarse en la RSA se obtiene la representación abstracta de la realidad, dada por los hechos (12) almacenados.

II.3 CRECIMIENTO DE LA RSA POR AGREGACION DE OBJETOS

Observando el crecimiento de la RSA, se ve que se da con la agregación de objetos y hechos. Un ejemplo de esto se ilustra en la figura 5.

Como se observa en 5d, al integrarse los objetos y los hechos se pueden obtener grupos separados.

Con el fin de observar las estructuras que se forman al agregarse los objetos se realizó un modelo para simular su integración, representando a un objeto con un pixel y los hechos con dos o más pixeles. La pantalla se considera como "la memoria" donde se almacena la información. Dado que se espera que los objetos lleguen en forma aleatoria y que se relacionen con los existentes de esta manera, se generan los pixeles en algún punto elegido al azar y se desplazan siguiendo un movimiento aleatorio (13), para que al llegar junto a otro se considere que tiene relación con él y por lo tanto se detenga formando un hecho.

(12) Un hecho se considera formado por dos objetos unidos por una relación (objeto1, relación, objeto2).

(13) En física a este tipo de movimiento se le conoce como Browniano en honor al físico Brown que lo descubrió en 1827 [19].

a) Llega el objeto "perro"

PERRO

b) Llega el hecho "un perro es un ser vivo"

PERRO

es un

SER VIVO

c) Se indica "un perro tiene cabeza"

CABEZA

tiene

PERRO

es un

SER VIVO

d) Se da "un oso es mamifero y come carne"

CABEZA

tiene

PERRO

es un

SER VIVO

OSO

es

MAMIFERO

come

CARNE

Figura 5.- Agregación de objetos en la RSA

Para simular los objetos que cuando llegan no tienen relación con otros se usa un indicador que establece aleatoriamente su aparición, éstos se detienen sin unirse con ningún otro. En el modelo se especifican dos parámetros:

- a) el número de pixeles
- b) el factor de densidad de agregación (FDA).

Con el **NUMERO DE PIXELES** se indica cuántos objetos se usan en la simulación.

El **FDA** especifica qué tan posible es el relacionamiento entre los objetos; si este parámetro es pequeño, se tiene poca posibilidad de relacionamiento, si es un número grande se considera que pueden quedar más relacionados.

Un ejemplo de lo que se puede obtener con este modelo se ilustra usando 500 pixeles variando el FDA, con valores de 100, 1000, 5000, 10000, 20000; los resultados se muestran en las figuras 6 a 10. En ellas se observa que a medida que se le indica al modelo que los pixeles tienen mayor posibilidad de quedar relacionados (aumenta el FDA) se obtienen cúmulos de mayor tamaño y menos objetos aislados.

Durante la formación de la red simulada, se observa que los pixeles se integran más rápidamente a medida que aumenta el número ellos.

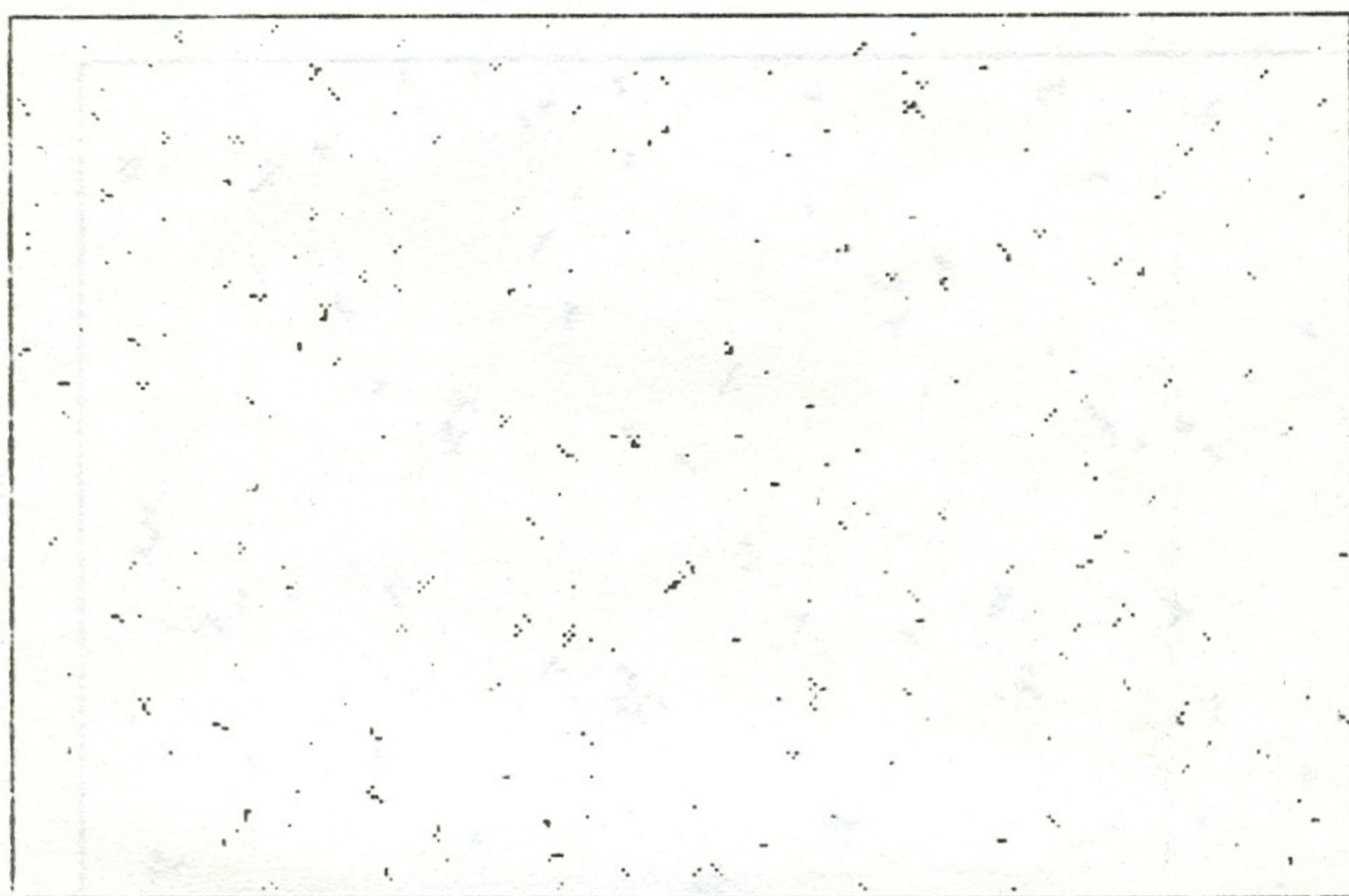


Figura 6.- PIXELES = 500, FDD = 100



Figura 7.- PIXELES = 500, FDA = 1000



Figura 8.- PIXELES = 500, FDA = 5000

Figura 9.- PIXELES = 500, FDA = 10000



Figura 9.- **FIXELES = 500, FDA = 10000**

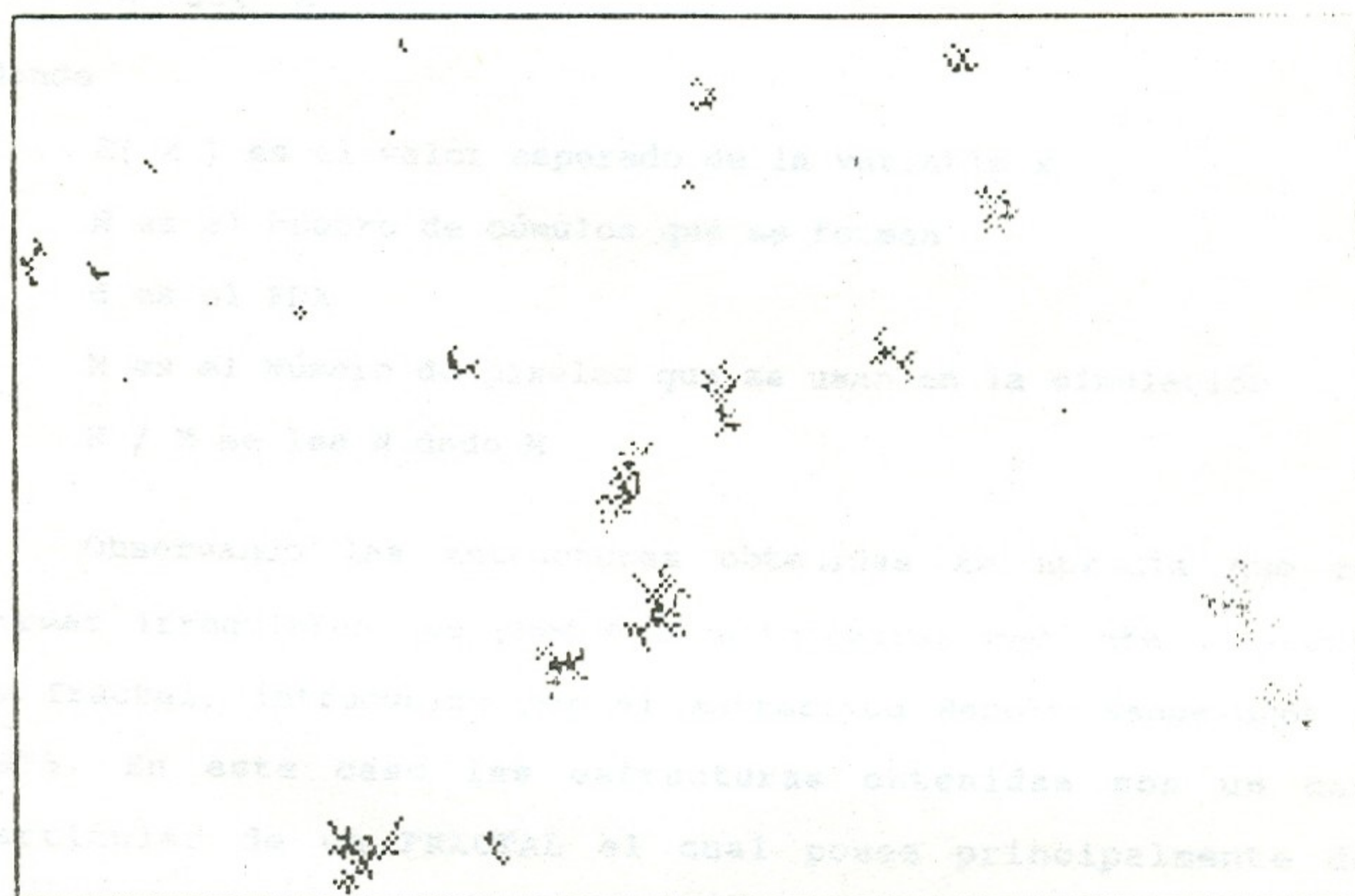


Figura 10.- $\text{PIXELES} = 500$, $\text{FDA} = 2000$

En este modelo una estimación empírica, suponiendo que existe suficiente área, el número de cúmulos que se pueden esperar variando el FDA es:

$$E(N / M) = 1$$

$$d \rightarrow \infty$$

y

$$E(N / M) = M$$

$$d \rightarrow 0$$

donde

$E(x)$ es el valor esperado de la variable x

N es el número de cúmulos que se forman

d es el FDA

M es el Número de píxeles que se usan en la simulación

N / M se lee N dado M

Observando las estructuras obtenidas se aprecia que son formas irregulares que pueden caracterizarse mediante la noción de fractal, introducido por el matemático Benoît Mandelbrot en 1975. En este caso las estructuras obtenidas son un caso particular de un FRACTAL el cual posee principalmente dos características [19] [20]:

- a) la autosimilaridad
- b) la dimensión fractal.

La AUTOSIMILARIDAD se puede apreciar en un objeto fractal cuando cada una de las partes del objeto fractal, cualesquiera que sean sus dimensiones es parecida o estadísticamente similar al todo. Esto se puede ilustrar con un árbol, al que se le corta

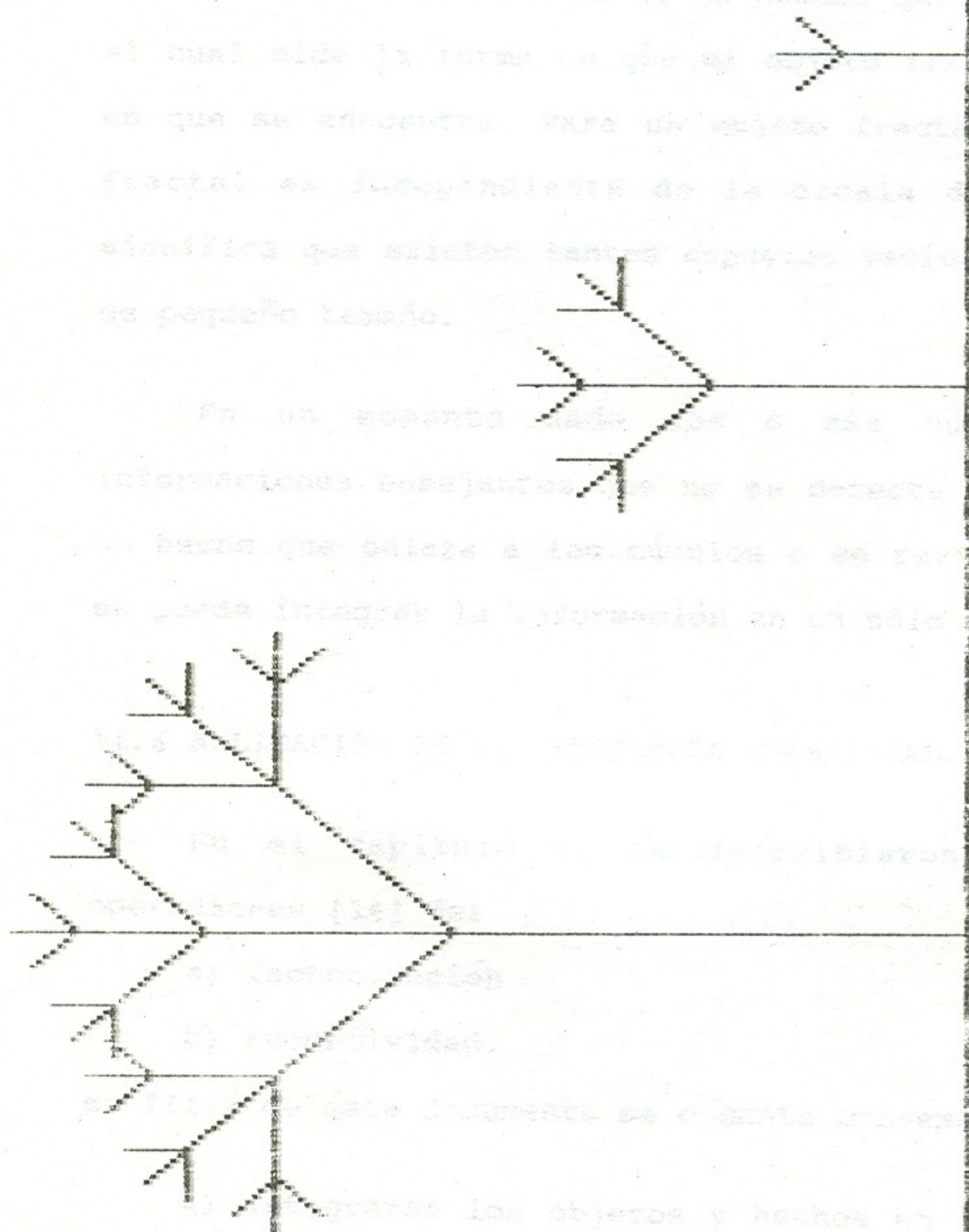


Figura 11.- Un ejemplo de objeto Fractal

una rama (una parte del todo), al plantarla en el suelo "se tiene la impresión de estar frente a un arbolito" [19] (figura 11). En el modelo un cúmulo se puede ver que está formado por cúmulos de menor tamaño, teniendo como componente básico un pixel.

La DIMENSION FRACTAL es un número que puede ser no entero, el cual mide la forma en que el objeto fractal llena el espacio en que se encuentra. Para un objeto fractal ideal la dimensión fractal es independiente de la escala de observación, esto significa que existen tantos espacios vacíos de gran tamaño como de pequeño tamaño.

En un momento dado dos o más cúmulos pueden tener informaciones semejantes que no se detecta sino hasta que llega un hecho que enlaza a los cúmulos o se revisa la red, dado esto se puede integrar la información en un sólo cúmulo (figura 12).

II.4 APLICACION DE LA INFERENCIA GRAMATICAL.

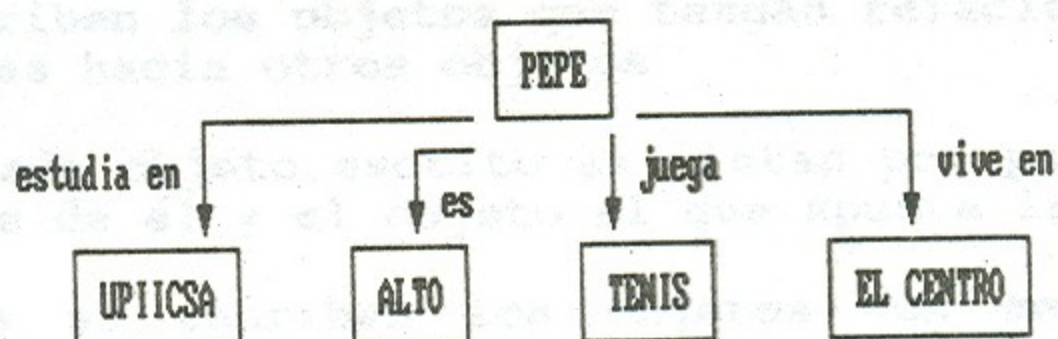
En el capítulo I, se describieron suscintamente las operaciones [14] de:

- a) factorización
- b) recursividad.

en III.2 de este documento se comenta brevemente la distribución.

Al integrarse los objetos y hechos en la RSA en un momento dado las relaciones hacia algunos objetos pueden estar repetidas, lo cual se puede aprovechar para obtener un factor común mediante la factorización y formar una regla del tipo:

a) Cúmulos separados



b) Después de la llegada del hecho "Pepe es Jose"

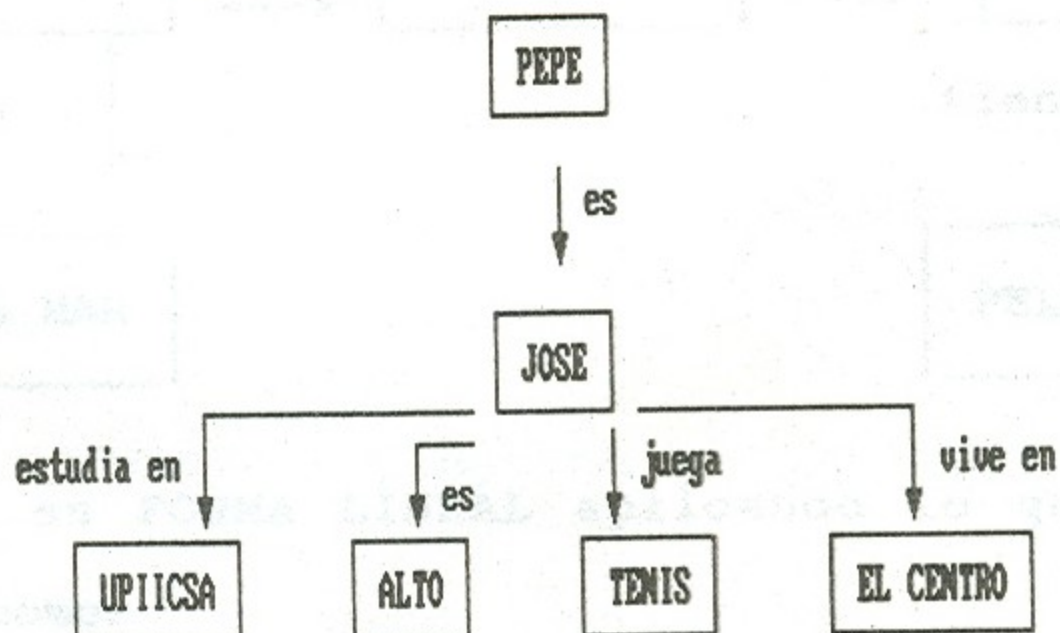


Figura 12.- Integración de información en la RSA

Si premisa ENTONCES conclusión

o proponer un objeto genérico al que se asignan los hechos comunes.

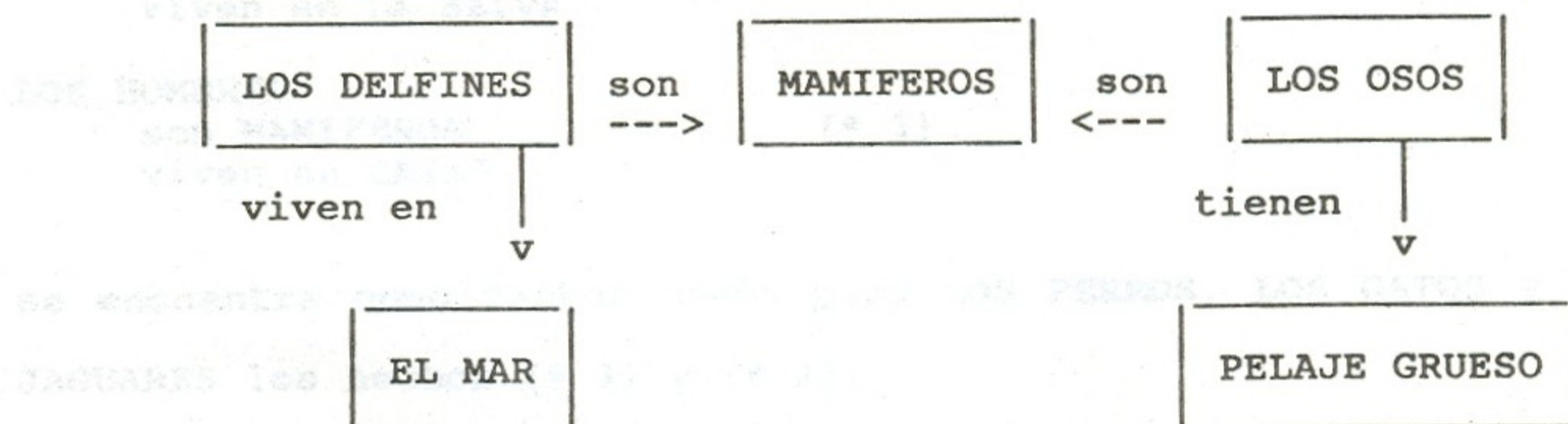
Para la aplicación de esta operación es útil ver la red en FORMA LINEAL que se obtiene de la siguiente forma:

- Se escriben los objetos que tengan relaciones con flechas dirigidas hacia otros objetos
- Para cada objeto escrito se listan por pares la relación que sale de él y el objeto al que apunta la relación.

De preferencia se escriben los objetos con mayúsculas y las relaciones con minúsculas para facilitar su lectura.

Ejemplo:

Considere la siguiente red.



al escribirla en FORMA LINEAL aplicando lo que se mencionó arriba, queda como:

LOS DELFINES
son MAMIFEROS
viven en EL MAR

LOS OSOS
tienen PELAJE GRUESO
son MAMIFEROS

En la RSA expresada en forma lineal se buscan hechos que se repitan un número suficiente de veces (por ejemplo tres) como para considerarlos que forman un factor común.

Ejemplo:

Considérese la RSA siguiente escrita en FORMA LINEAL.

LOS DELFINES
son MAMIFEROS (* 1)
viven en EL MAR

LOS PERROS
son MAMIFEROS (* 1)
respiran por LOS PULMONES (* 2)
comen CARNE

LOS GATOS
son MAMIFEROS (* 1)
respiran por LOS PULMONES (* 2)
comen CARNE

LOS JAGUARES
son MAMIFEROS (* 1)
respiran por LOS PULMONES (* 2)
viven en LA SELVA

LOS HOMBRES
son MAMIFEROS (* 1)
viven en CASAS

se encuentra como factor común para LOS PERROS, LOS GATOS y LOS JAGUARES los hechos (* 1) y (* 2):

"son MAMIFEROS" y "respiran por LOS PULMONES"

de este factor común se puede proponer una regla de la forma mencionada, que sirve para algunos sistemas expertos [26]:

SI X son MAMIFEROS ENTONCES respiran por LOS PULMONES

Formas más elaboradas de ésta pueden obtenerse introduciendo los operadores lógicos "Y", "O", "NO".

Otra forma en que se puede usar lo que se obtuvo es estableciendo un objeto genérico, con él se relaciona cada caso particular y en éstos se eliminan los hechos comunes. Cada objeto particular conserva sus hechos de manera indirecta por su relación con el genérico.

Proponiendo a "MAMIFEROS" como objeto genérico, el ejemplo anterior se expresa como sigue:

LOS DELFINES
son MAMIFEROS
viven en EL MAR

LOS PERROS
comen CARNE
son MAMIFEROS

LOS GATOS
comen CARNE
son MAMIFEROS

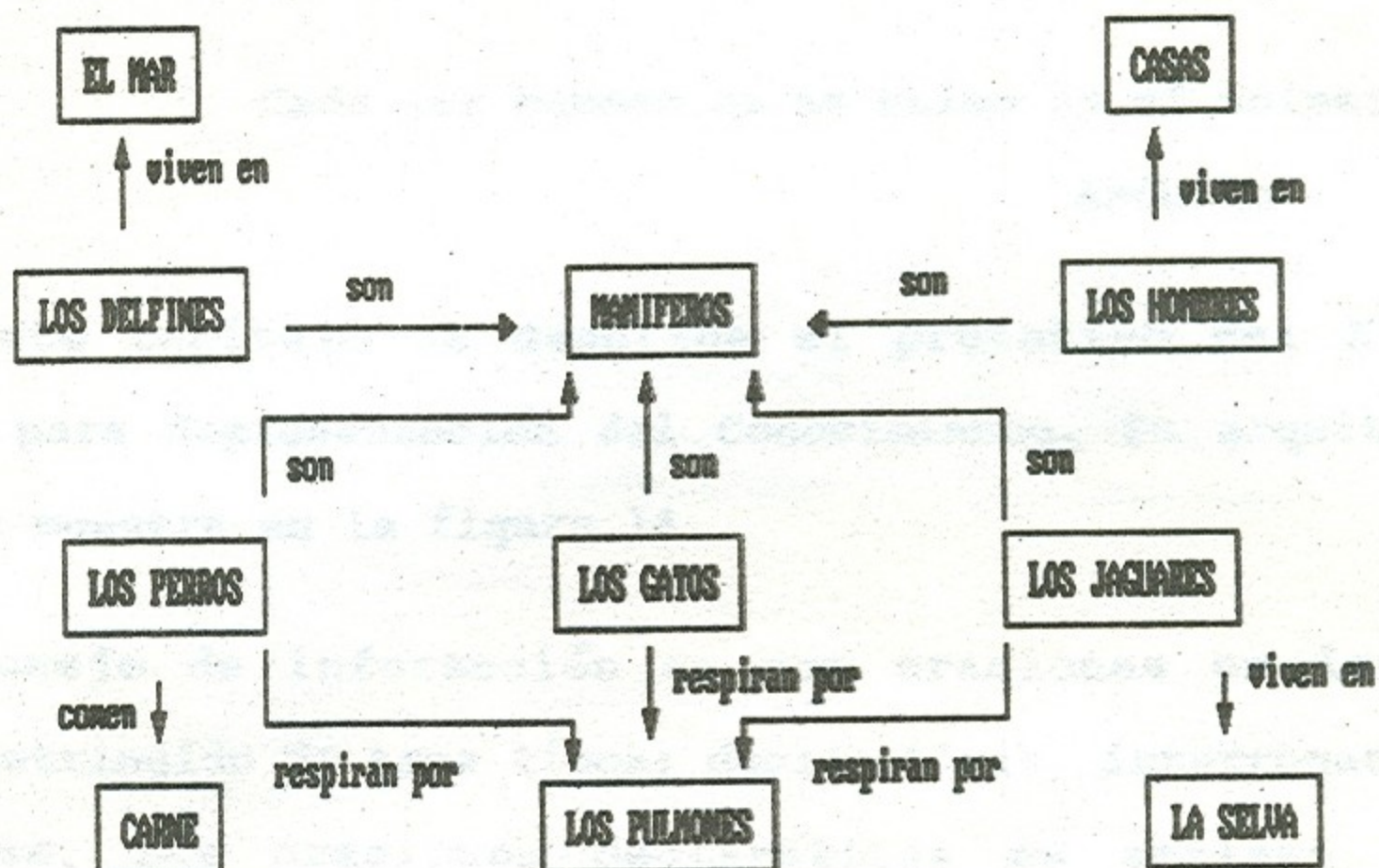
LOS JAGUARES
viven en SELVA
son MAMIFEROS

LOS HOMBRES
son MAMIFEROS
viven en CASAS

MAMIFEROS
respiran por LOS PULMONES

aquí se observa que los objetos que tienen relación con "MAMIFEROS" adquieren automáticamente sus propiedades, en este caso "LOS DELFINES" y "LOS HOMBRES" se consideran con "respiran por LOS PULMONES" a través de que son "MAMIFEROS" (figura 13).

a) RSA antes de factorizar



b) Después de asignar a MANIFEROS como objeto genérico

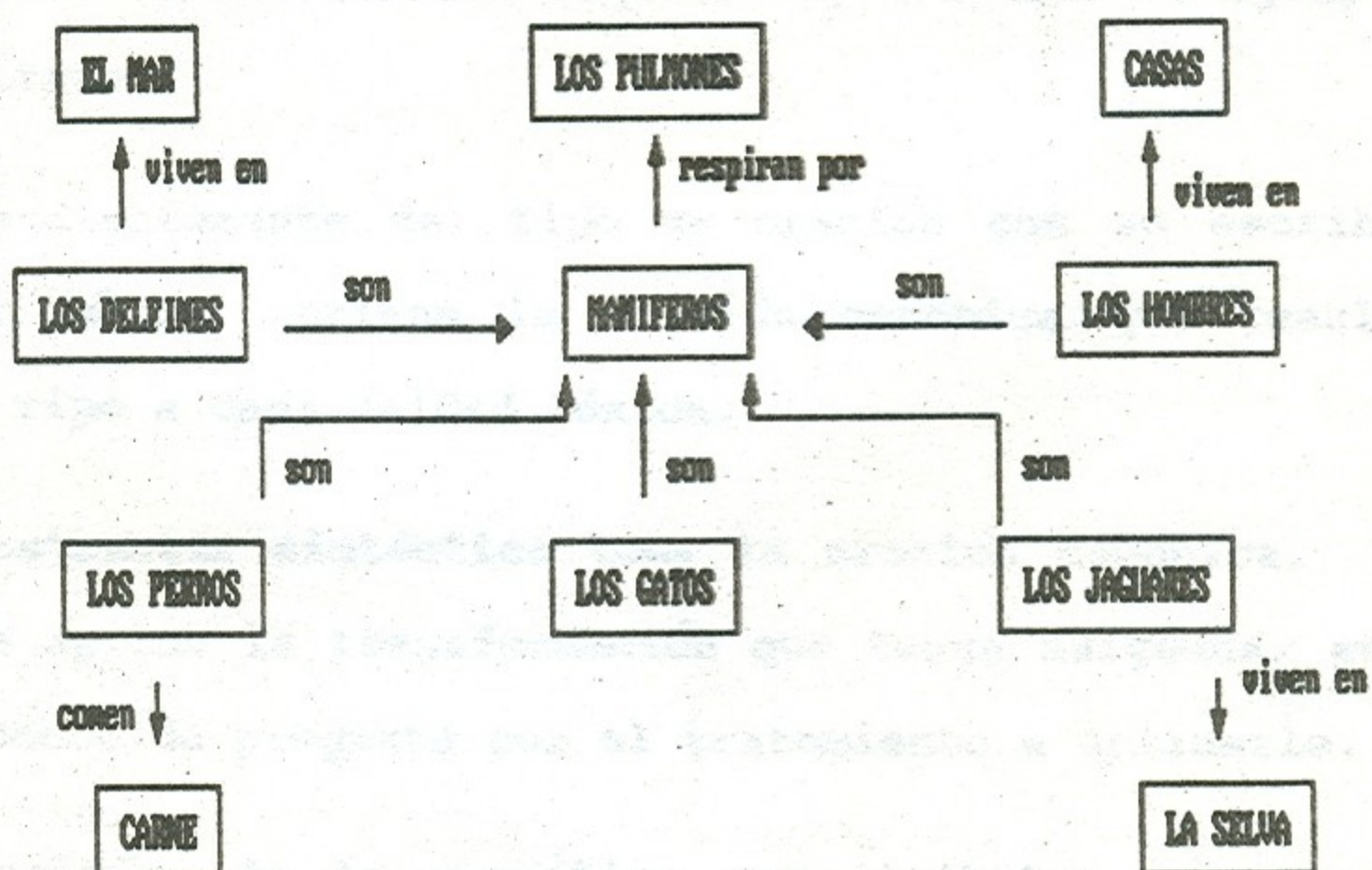


Figura 13.- Aplicación de la factorización en la RSA

CAPITULO III. DESCRIPCION DEL PROTOTIPO.

Cada ser humano es un único en el universo.

Anónimo

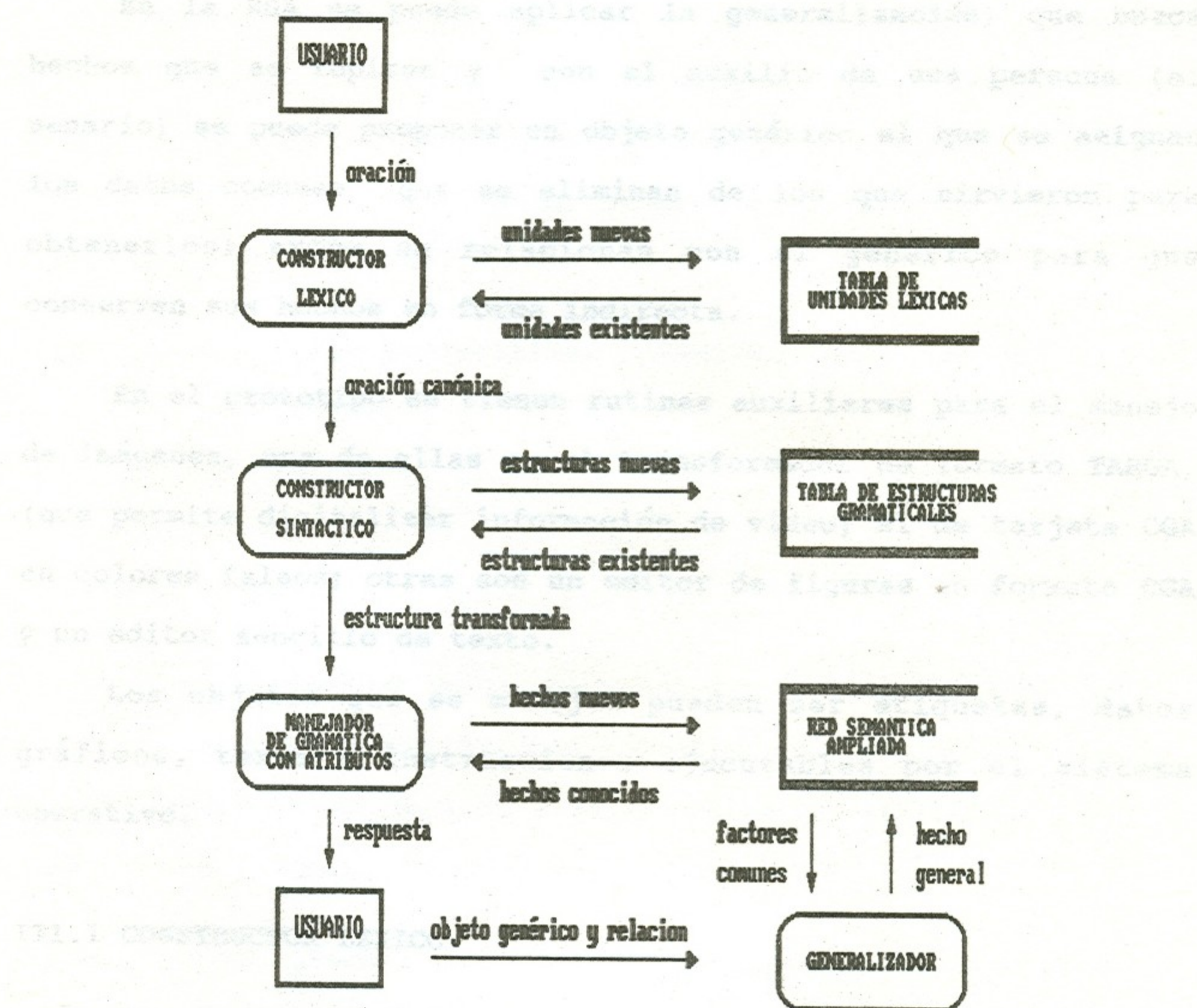
En este capítulo se describe el prototipo del Sistema Evolutivo para Representación del Conocimiento. Su arquitectura general se muestra en la figura 14.

El manejo de información es con oraciones en lenguaje natural restringido de tres tipos: declarativas, interrogativas e imperativas. Las oraciones declarativas se emplean en el constructor para obtener los hechos que se integran a la Red Semántica Ampliada. Las de tipo interrogativo e imperativo sirven para consultar la información registrada, con esto se ejemplifica el Administrador.

Independientemente del tipo de oración que se escriba, el constructor léxico obtiene la oración canónica que resulta de asignar un tipo a cada unidad léxica.

El constructor sintáctico toma la oración canónica, si es conocida le aplica la transformación que tenga asignada, en caso de ser desconocida pregunta por el tratamiento a aplicarle.

El manejador de la gramática con atributos interpreta la oración canónica transformada y como resultado integra los hechos en la red o genera una respuesta; según el tipo de oración que se haya escrito.



SIMBOLOGIA:



Figura 14.- Diagrama de Flujo de Datos del prototipo

En la RSA se puede aplicar la **generalización**, que busca hechos que se repiten y con el auxilio de una persona (el usuario) se puede proponer un objeto genérico al que se asignan los datos comunes que se eliminan de los que sirvieron para obtenerlos; estos se relacionan con el genérico para que conserven sus hechos en forma indirecta.

En el prototipo se tienen rutinas auxiliares para el manejo de imágenes, una de ellas es el transformador de formato TARGA, (que permite digitalizar información de video) al de tarjeta CGA en colores falsos; otras son un editor de figuras en formato CGA y un editor sencillo de texto.

Los objetos que se manejan pueden ser etiquetas, datos gráficos, texto e instrucciones ejecutables por el sistema operativo.

III.1 CONSTRUCTOR LEXICO.

Este módulo obtiene las oraciones canónicas de las oraciones que se dan al prototipo, estas pueden ser declarativas, interrogativas e imperativas y de ellas se le indican las unidades léxicas que son nuevas.

En su proceso de identificación de las unidades, se recorre la oración carácter por carácter, en cada uno se busca en la tabla de unidades léxicas alguna que inicie con ese carácter y que corresponda en la oración; cuando la encuentra salta los caracteres de la misma.

Si después de examinar la oración se detecta que hay uno o varios caracteres sin reconocer, se pregunta por alguna unidad léxica que este entre lo que se desconoce. La nueva unidad se marca en la oración y el usuario le asigna su atributo que puede ser su tipo o un sinónimo.

En las oraciones declarativas interesan los tipos que sirven para formar los hechos:

a) objetos (o, f, t, s)

b) relaciones (r)

c) ignorable (i).

El tipo "ignorable" sirve para asignarlo a aquello que no es relevante para obtener hechos. Cuando un tipo no existe, automáticamente se da de alta y se pregunta al usuario el nombre que le quiera asignar, éste sólo es informativo.

Los tipos pueden ser letras minúsculas o caracteres especiales.

Los sinónimos se identifican con un número que debe indicarse en el atributo de la unidad léxica. El propósito de manejar los sinónimos permite que dos o más unidades que tengan significado similar se consideren de esta manera. Por ejemplo si algunas relaciones son semejantes y varían por su flexión verbal, se pueden asignar como sinónimos escribiendo en su atributo el mismo número, como "brinca", "brincan", "brincamos" que haciendo esto aparecen como sigue:

TABLA DE UNIDADES LEXICAS

<u>Número</u>	<u>Unidad Léxica</u>	<u>Tipo</u>	<u>Sinónimo</u>
1	LOS	i	1
2	SERES VIVOS	i	2
3	BRINCA	r	3
4	ANTENAS	o	4
5	BRINCAN	r	3
6	AVISPAS	o	6
7	BRINCAMOS	r	3

Después de asignar su tipo a cada unidad léxica se obtiene la oración canónica, de ésta se eliminan las que lleven tipo ignorable. Luego se numeran los objetos y las relaciones de la canónica de manera independiente, cada numeración inicia en uno.

Se muestra a continuación un ejemplo de las funciones de este constructor aplicado a oraciones que se usaran en los sucesivo para ejemplificar las rutinas mas importantes.

Cada ejemplo lleva un número y tres incisos: el a, es la oración que sera examinada; el b, es el resultado de asignar un número de unidad y su tipo, este valor puede ser de un sinónimo; en c, se tiene el resultado de quitar los ignorables y numerar los objetos y relaciones. Su tabla correspondiente aparece al final de ellas.

1.- a) LA BIOLOGIA	ESTUDIA A	LOS SERES VIVOS
└	└	└
v	v	v
b) lo	2i	3r
└	└	└
v	v	v
c) ol	rl	o2

III.2 CONSTRUCTOR SINTACTICO

Este módulo toma la oración canónica que se obtuvo anteriormente y le aplica su transformación asociada que permite obtener explícitos los hechos en forma de oraciones sencillas (o1 r o2) separadas por ";" a partir de oraciones en que los hechos no están así.

La primera ocasión que se ejecuta el módulo no tiene ninguna estructura. Cada vez que llega una nueva se pregunta al usuario por la transformación a aplicar y que puede ser:

- a) usar la original.- la estructura obtenida permanece igual
- b) asignar sinónimo.- en este caso el usuario indica la forma en que se deben distribuir los componentes de la oración canónica
- c) distribución [23].- se aplica esta operación y se obtienen oraciones sencillas

Después de distribuir la oración canónica sus elementos quedan reasignados de acuerdo con lo que se haya indicado.

La DISTRIBUCION de la Inferencia Gramatical se basa en la distribución de expresiones algebraicas, de la que se muestra un ejemplo:

De la expresion algebraica

$$(a + b) (c + d + e)$$

al distribuir los términos, se obtiene

$$a (c + d + e) + b (c + d + e)$$

$$a c + a d + a e + b c + b d + b e$$

2.- a) LOS PERROS Y LOS GATOS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE

b) 5o 2i 6i 2i 7o 2i 8r 2i 9o 10i 2i 11r 2i 12o
2i 6i 2i 13r 2i 14o

c) o1 o2 r1 o3 r2 o4 r3 o5

3.- a) LOS SERES VIVOS SE DIVIDEN EN VEGETALES Y ANIMALES

b) 4o 2i 15r 2i 16o 2i 6i 2i 17o

c) o1 r1 o2 o3

4.- a) LOS GATOS Y LOS HOMBRES SON MAMIFEROS QUE VIVEN EN LAS CASAS

b) 7o 2i 6i 2i 18o 2i 8r 2i 9o 19i 20r 2i 21o

c) o1 o2 r1 o3 r2 o4

5.- a) LOS JAGUALES TIENEN RESPIRACION PULMONAR, SON MAMIFEROS Y VIVEN EN LA SELVA

b) 22o 2i 11r 2i 12o 2i 10i 2i 8r 2i 9o 2i 6i 2i
20r 2i 25o

c) o1 r1 o2 r2 o3 r3 o4

6.- a) LOS DELFINES SON MAMIFEROS QUE VIVEN EN EL MAR

b) 26o 2i 8r 2i 9o 2i 19i 2i 20r 2i 27o

c) o1 r1 o2 r2 o3

7.- a) LOS VEGETALES TOMAN Y FRACCIONAN EL BIOXIDO DE CARBONO

b) 16o 2i 29r 2i 6i 2i 30r 2i 31o

c) o1 r1 r2 o2

TABLA DE UNIDADES LEXICAS

<u>Número</u>	<u>Unidad Léxica</u>	<u>Tipo</u>	<u>Sinónimo</u>
1	LA BIOLOGIA	o	1
2		i	2
3	ESTUDIA A	r	3
4	LOS SERES VIVOS	o	4
5	LOS PERROS	i	5
6	Y	i	6
7	LOS GATOS	o	7
8	SON	r	8
9	MAMIFEROS	o	9
10	,	i	10
11	RESPIRAN POR	r	11
12	LOS PULMONES	o	12
13	COMEN	r	13
14	CARNE	o	14
15	SE DIVIDEN EN	r	15
16	VEGETALES	o	16
17	ANIMALES	o	17
18	LOS HOMBRES	o	18
19	QUE	i	19
20	VIVEN EN	r	20
21	LAS CASAS	o	21
22	LOS JAGUARES	o	22
23	TIENEN RESPIRACION	r	11
24	PULMONAR	o	12
25	LA SELVA	o	25
26	LOS DELFINES	o	26
27	EL MAR	o	27
28	LOS VEGETALES	o	28
29	TOMAN	r	29
30	FRACCIONAN	r	30
33	EL BIOXIDO DE CARBONO	o	31

lo que se comprueba aritméticamente considerando

$$a = 3, b = 4, c = 1, d = 2, e = 5$$

$$\begin{aligned} (3 + 4) (1 + 2 + 5) &= 3 (1 + 2 + 5) + 4 (1 + 2 + 5) \\ (7) (8) &= 3*1 + 3*2 + 3*5 + 4*1 + 4*2 + 4*5 \\ 7*8 &= 3 + 6 + 15 + 4 + 8 + 20 \\ 56 &= 56 \end{aligned}$$

Los pasos de la DISTRIBUCION LINGUISTICA son:

- a) Se toma la oración canónica
- b) Se aplica la **Distribucion Cero** que consiste en dejar en un primer paréntesis los objetos que aparecen al inicio de la oración canónica separados con el operador más (+). En un segundo paréntesis se asignan por pares las relaciones y objetos que aparecen a continuación de los primeros objetos. Cada par se separa con el operador más (+)
- c) Al resultado se le aplica la **Primera Distribución** asignando cada objeto del primer paréntesis con el segundo paréntesis
- d) A lo que resulte se le aplica la **Segunda Distribución** mediante la que se eliminan los paréntesis, asignando cada objeto fuera del paréntesis con cada par relación-objeto
- e) Por último la **Adecuacion** consiste en sustituir los operadores más (+) por separadores de oraciones punto y coma (;) y anexar un terminador de oración (un punto, ".").

La aplicación de la DISTRIBUCION LINGUISTICA a la oración

LOS	PERROS	Y	LOS	GATOS	SON	MAMIFEROS,	RESPIRAN	POR	LOS	PULMONES	Y	
	o1			o2		r1		o3		r2		o4
COMEN	CARNE											
	r3			o5								

es como sigue:

- a) Se toma la oración canónica:

o1 o2 r1 o3 r2 o4 r3 o5

b) Se aplica la Distribución Cero:

$$(o1 + o2) (r1 o3 + r2 o4 + r3 o5)$$

c) Se aplica la Primera Distribución:

$$o1 (r1 o3 + r2 o4 + r3 o5) + o2 (r1 o3 + r2 o4 + r3 o5)$$

d) Se aplica la Segunda Distribución:

$$o1 r1 o3 + o1 r2 o4 + o1 r3 o5 + o2 r2 o3 + o2 r2 o4 + o2 r3 o5$$

e) Se adecua la oración canónica obtenida:

$$o1 r1 o3 ; o1 r2 o4 ; o1 r3 o5 ; o2 r2 o3 ; o2 r2 o4 ; o2 r3 o5 .$$

Con la oración canónica obtenida se reasignan las unidades léxicas de la oración original para obtener hechos que se registran directamente en la RSA, mismos que son:

LOS PERROS o1	SON r1	MAMIFEROS o3	;
LOS PERROS o1	RESPIRAN POR r2	LOS PULMONES o4	;
LOS PERROS o1	COMEN r3	CARNE o5	;
LOS GATOS o2	SON r1	MAMIFEROS o3	;
LOS GATOS o2	RESPIRAN POR r2	LOS PULMONES o4	;
LOS GATOS o2	COMEN r3	CARNE o5	.

Al aplicar la distribución a las oraciones que se manejan como ejemplo de los módulos se obtiene lo que está más abajo. El inciso a, es la oración original; el b, es la oración canónica obtenida por el constructor léxico; el c, muestra la oración canónica distribuida; y el d, tiene el resultado de la distribución.

1.- a) LA BIOLOGIA ESTUDIA A LOS SERES VIVOS

b) o1 r1 o2

c) o1 r1 o2

d) LA BIOLOGIA estudia a LOS SERES VIVOS

2.- a) LOS PERROS Y LOS GATOS SON MAMIFEROS, RESPIRAN POR LOS PULMONES Y COMEN CARNE

b) o1 o2 r1 o3 r2 o4 r3 o5

c) o1 r1 o3 ; o1 r2 o4 ; o1 r3 o5 ; o2 r1 o3 ;
o2 r2 o4 ; o2 r3 o5 .

d) LOS PERROS son MAMIFEROS ;
LOS PERROS respiran por LOS PULMONES ;
LOS PERROS comen CARNE ;
LOS GATOS son MAMIFEROS ;
LOS GATOS respiran por LOS PULMONES ;
LOS GATOS comen CARNE .

3.- a) LOS SERES VIVOS SE DIVIDEN EN VEGETALES Y ANIMALES

b) o1 r1 o2 o3

c) o1 r1 o2 ; o1 r1 o3 .

d) LOS SERES VIVOS se dividen en VEGETALES ;
LOS SERES VIVOS se dividen en ANIMALES .

4.- a) LOS GATOS Y LOS HOMBRES SON MAMIFEROS QUE VIVEN EN LAS CASAS

b) o1 o2 r1 o3 r2 o4

c) o1 r1 o3 ; o1 r2 o4 ; o2 r1 o3 ; o2 r2 o4 .

d) LOS GATOS son MAMIFEROS ;
LOS GATOS viven en LAS CASAS ;
LOS HOMBRES son MAMIFEROS ;
LOS HOMBRES viven en LAS CASAS .

5.- a) LOS JAGUALES TIENEN RESPIRACION PULMONAR, SON MAMIFEROS Y VIVEN EN LA SELVA

b) o1 r1 o2 r2 o3 r3 o4

c) o1 r1 o2 ; o1 r2 o3 ; o1 r3 o4 .

- d) LOS JAGUARES respiran por LOS PULMONES ;
 LOS JAGUARES son MAMIFEROS ;
 LOS JAGUARES viven en LA SELVA .

6.- a) LOS DELFINES SON MAMIFEROS QUE VIVEN EN EL MAR

b) o1 r1 o2 r2 o3

c) o1 r1 o2 ; o1 r2 o3 .

- d) LOS DELFINES son MAMIFEROS ;
 LOS DELFINES viven en EL MAR .

7.- a) LOS VEGETALES TOMAN Y FRACCIONAN EL BIOXIDO DE CARBONO

b) o1 r1 r2 o2

c) o1 r1 o2 ; o1 r2 o2 .

- d) LOS VEGETALES toman EL BIOXIDO DE CARBONO ;
 LOS VEGETALES fraccionan BIOXIDO DE CARBONO .

El algoritmo para realizar la distribución es el siguiente:

VARIABLES:

Oracion canónica, llamada "la oración";
 áreal, área de trabajo;
 área2, área de trabajo;
 área3, es donde queda el resultado de la distribución;
 lista de objetos, llamada "L. objetos";
 lista de relaciones, llamada "L. relaciones".

1.- Pon paréntesis, "(", en areal

2.- operador = '+'

3.- Mientras haya objetos en la oración

3.1.- pon en áreal cada objeto de la oración separado del siguiente con el operador

4.- Pon paréntesis, ")" y "(", en areal

5.- Mientras haya símbolos en la oración (DISTRIBUCION CERO)

5.1.- Inicializa L. objetos y L. relaciones

5.2.- Mientras haya relaciones en la oración

5.2.1.- guarda la relación en L. relaciones

5.3.- Mientras haya objetos en la oración

5.3.1.- guarda el objeto en L. objetos

5.4.- Pon en áreal todos los pares posibles relación-objeto de L. relaciones contra L. objetos separados con el operador

6.- Pon paréntesis, ")" en áreal

7.- Para cada objeto del primer paréntesis de áreal (PRIMERA DISTRIBUCION)

7.1.- Pasa cada objeto y el segundo paréntesis a area2 separando cada par con el operador

8.- Para cada objeto-paréntesis de area2 (SEGUNDA DISTRIBUCION)

8.1.- Asigna cada objeto con cada par relacion-objeto dentro del paréntesis pasando de área2 a área3; cada asignación se separa con el operador

9.- Se sustituye el operador por ";" y se anexa un punto al final de área3 (ADECUACION).

III.3 MANEJADOR DE GRAMATICA CON ATRIBUTOS.

Este módulo interpreta las oraciones transformadas del constructor sintáctico con una gramática con atributos que permite reconocer las oraciones que se dan al sistema.

La gramática con atributos que se utiliza para interpretar las oraciones declarativas es la siguiente:

$$GA = \{ V_n, V_t, V_s, P, A \}$$

$$V_n = \{ A, B, O \}$$

$$V_t = \{ o, f, r, s, t, ;, . \}$$

$$V_s = \{ 1 \}$$

$$P = \{$$

A	->	O	r	O	1	B
B	->	;	O	r	O	1 B
B	->	.				
O	->	o				
O	->	f				
O	->	t				
O	->	s				

$$\}$$

Si alguna oración no cumple con esta gramática se indica al usuario con un aviso. En caso de requerirse se puede modificar la gramática con atributos para que reconozca otras oraciones o indicar alguna transformación en el constructor sintáctico para este propósito.

La rutina semántica de la gramática anterior que permite integrar hechos a la RSA es:

Rutina 1.- Se busca en la RSA el primer objeto y se deja ubicado como objeto 1, en caso que no exista, se da de alta. Se toma la relación. Se busca el segundo objeto en la RSA y se deja ubicado como objeto 2, en caso de no existir se da de alta.

Si la oración es válida para cada hecho se busca la relación y el objeto 2 en los hechos del objeto 1 y si no están los anexa al objeto 1; Si existen no se consideran.

Al interpretar la oraciones que se vienen manejando como ejemplo se obtiene la figura 15.

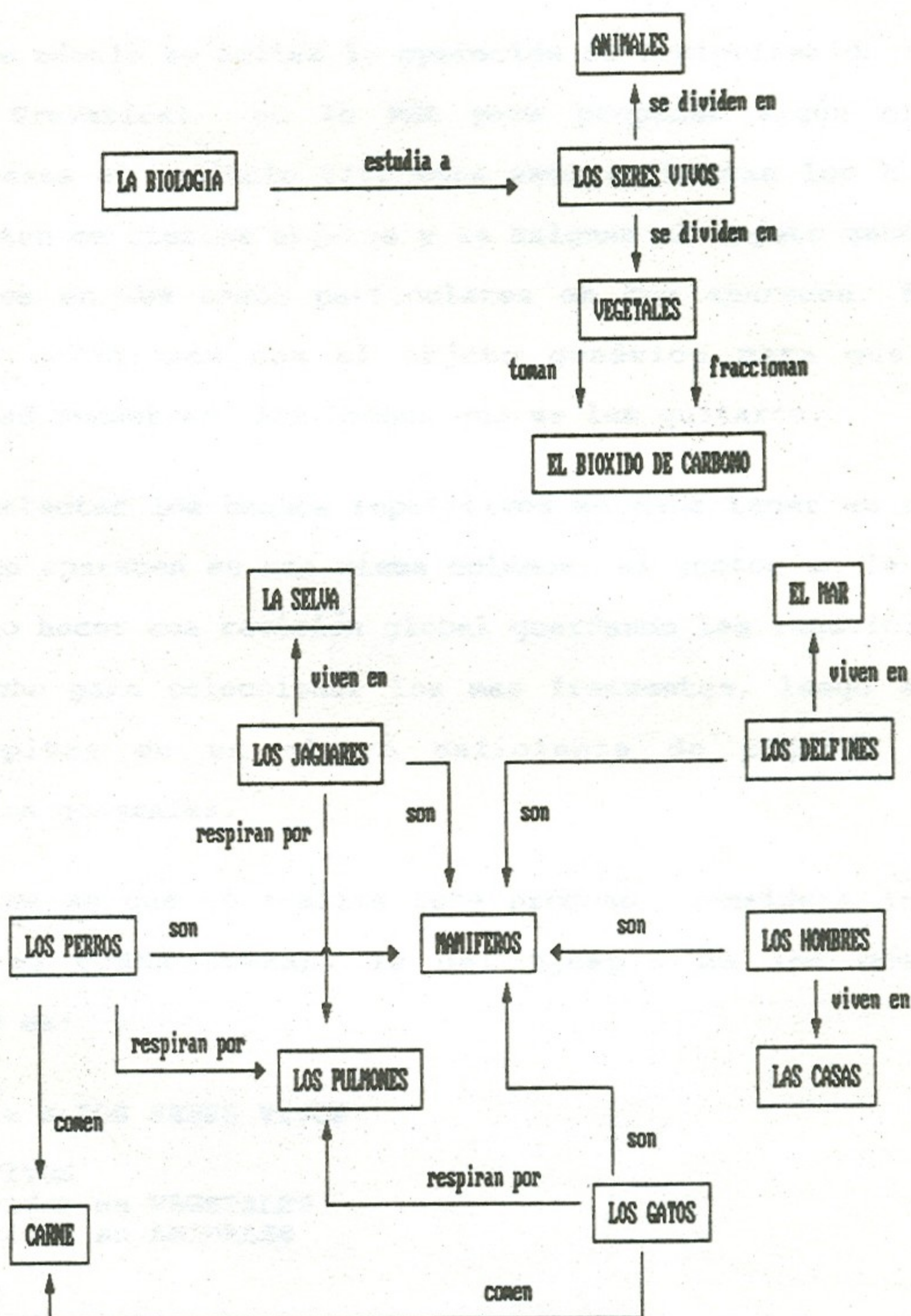


Figura 15.- Red Semántica Ampliada de las oraciones de ejemplo

III.4 GENERALIZADOR

En este módulo se aplica la operación de Factorización de la Inferencia Gramatical en la RSA para proponer algún objeto genérico (véase el capítulo II). Para esto se buscan los hechos que se repiten en ciertos objetos y se asignan al objeto genérico eliminándolos en los casos particulares en que aparecen. Estos últimos se relacionan con el objeto genérico para que por transitividad conserven los hechos que se les quitaron.

Para detectar los hechos repetitivos se debe tener en mente que estos no aparecen en una misma columna, ni juntos en la RSA; es necesario hacer una revisión global guardando las repeticiones de cada hecho para seleccionar los mas frecuentes, luego se ve que se repitan en un número suficiente de objetos para considerarlos generales.

La forma en que se realiza este proceso considera la RSA expresada en FORMA LINEAL, la del ejemplo de los módulos principales es:

LA BIOLOGIA
estudia a LOS SERES VIVOS

LOS SERES VIVOS
se dividen en VEGETALES
se dividen en ANIMALES

LOS PERROS
son MAMIFEROS
respiran por LOS PULMONES
comen CARNE

LOS GATOS
son MAMIFEROS
respiran por LOS PULMONES
comen CARNE
viven en LAS CASAS

VEGETALES

toman EL BIOXIDO DE CARBONO
fraccionan EL BIOXIDO DE CARBONO

LOS HOMBRES

son MAMIFEROS
viven en LAS CASAS

LOS JAGUARES

respiran por LOS PULMONES
son MAMIFEROS
viven en LA SELVA

LOS DELFINES

son MAMIFEROS
viven en EL MAR

Los hechos que aparecen un número suficiente de veces (14),
por ejemplo tres son:

<u>HECHO</u>	<u>OBJETOS EN DONDE APARECE</u>
son MAMIFEROS	LOS PERROS, LOS GATOS, LOS HOMBRES, LOS JAGUARES, LOS DELFINES
respiran por LOS PULMONES	LOS PERROS, LOS GATOS, LOS JAGUARES

De estos interesa que tomándolos por pares, aparezcan en un
número suficiente de objetos (p. ej. tres veces):

<u>HECHOS APAREADOS</u>	<u>OBJETOS EN QUE APARECEN</u>
1) son MAMIFEROS Y respiran por LOS PULMONES	LOS PERROS, LOS GATOS, LOS JAGUARES

De estos se selecciona uno de los pares y sus hechos se le
asignan al objeto que se denomine como genérico.

Si se selecciona "MAMIFEROS" como genérico, se observa que
este aparece en los hechos generales, por lo tanto, sólo se
considera el que no tiene a este objeto.

(14) Aquí se utiliza tres, pero puede ser otro valor.

Después de asignar el objeto genérico "MAMIFEROS" y relacionar los casos particulares con él se obtiene:

LA BIOLOGIA
estudia a LOS SERES VIVOS

LOS SERES VIVOS
se dividen en VEGETALES
se dividen en ANIMALES

LOS PERROS
comen CARNE
son MAMIFEROS

LOS GATOS
comen CARNE
viven en LAS CASAS
son MAMIFEROS

VEGETALES
toman EL BIOXIDO DE CARBONO
fraccionan EL BIOXIDO DE CARBONO

LOS HOMBRES
viven en LAS CASAS
son MAMIFEROS

LOS JAGUARES
viven en LA SELVA
son MAMIFEROS

LOS DELFINES
son MAMIFEROS
viven en EL MAR

MAMIFEROS
respiran por LOS PULMONES

Como se mencionó en el capítulo II, algunos objetos al estar relacionados con el objeto que se hizo genérico adquieren automáticamente sus hechos en forma indirecta. En este ejemplo "LOS HOMBRES" y "LOS DELFINES" tienen "respiran por LOS PULMONES" por estar relacionados con "MAMIFEROS".

Aplicando la factorización a los ejemplos de los módulos se obtiene la figura 16.

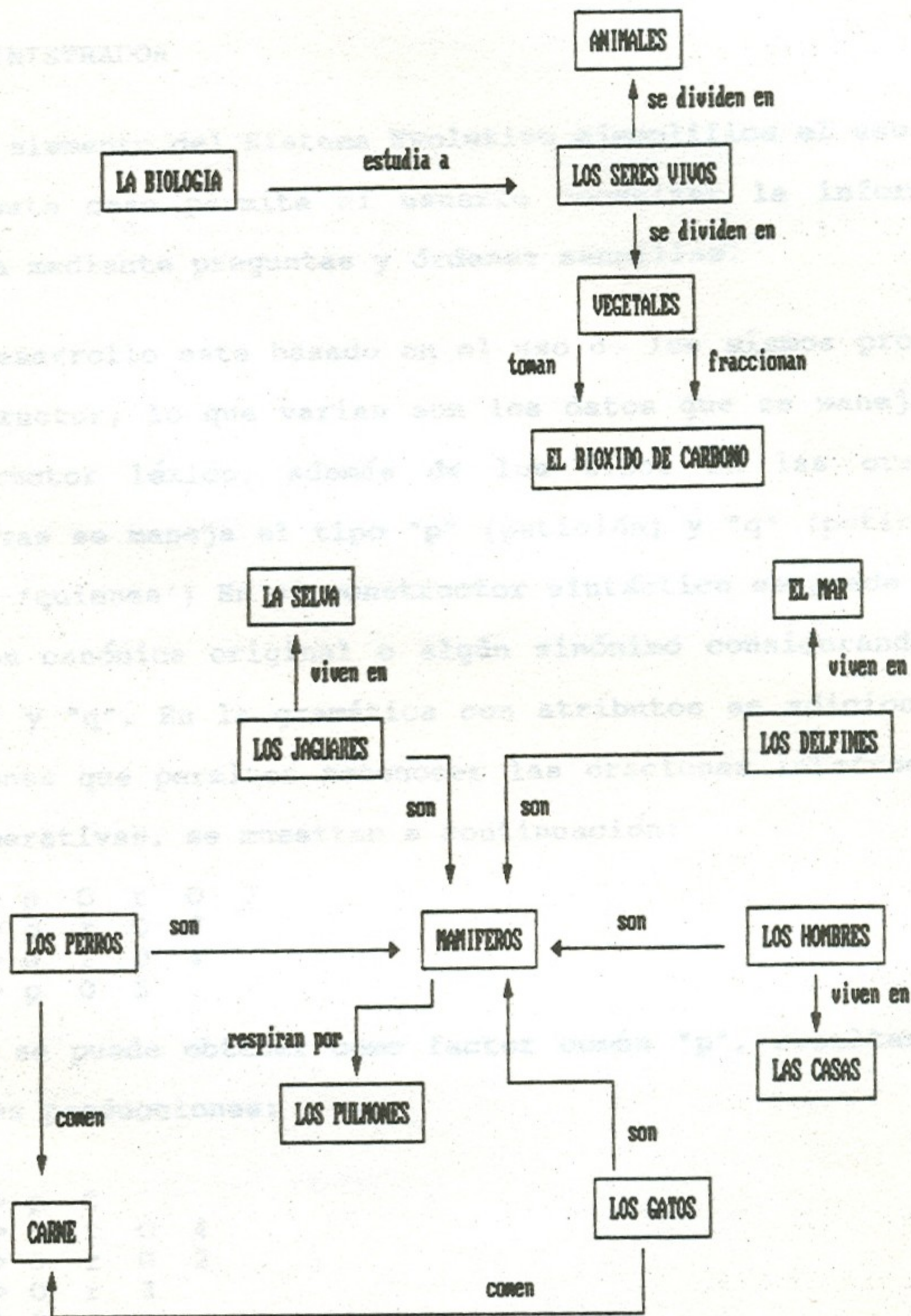


Figura 16.- Red Semántica Ampliada del ejemplo después de aplicar Factorización

III.5 ADMINISTRADOR

Este elemento del Sistema Evolutivo ejemplifica el uso de la RSA, en este caso permite al usuario consultar la información registrada mediante preguntas y órdenes sencillas.

Su desarrollo esta basado en el uso de los mismos programas del constructor, lo que varían son los datos que se manejan. En el constructor léxico, además de los tipos de las oraciones declarativas se maneja el tipo "p" (petición) y "q" (petición de 'quien' o 'quienes') En el constructor sintáctico se puede asignar la oración canónica original o algún sinónimo considerando los tipos "p" y "q". En la gramática con atributos se adicionan las producciones que permiten reconocer las oraciones interrogativas y las imperativas, se muestran a continuación:

```
A -> p 0 r 0 2
A -> p r 0 3
A -> q r 0 4
A -> p 0 5
```

en estas se puede obtener como factor común "p", resultando las siguientes producciones:

```
A -> p C
A -> q r 0 4
C -> 0 r 0 2
C -> 0 r 3
C -> 0 5
```

Las rutinas semánticas que aparecen en las producciones nuevas realizan lo siguiente:

Rutina 2: Se toma el hecho formado con los objetos y la relación que le preceden, se busca en la RSA, si existe se indica "CIERTO", de lo contrario se indica "NO TENGO ESTE DATO".

Rutina 3: Se toman la relación y el objeto. Se indican los que se relacionen desde el que se dá, mediante la relación.

Rutina 4: Se busca en la RSA los objetos que tengan asociada directa o indirectamente la relación y objetos dados.

Rutina 5: Se indican todos los hechos que se tengan en la RSA en forma directa o indirecta para el objeto dado.

Las producciones anteriores resultan de las oraciones sencillas con que puede ser consultado el prototipo. Algunos ejemplos de ellas se muestran a continuación. Para cada ejemplo se tienen varios incisos: el a, es la oración que da el usuario; el b, es la oración canónica que obtiene el constructor léxico; el c, es la transformada; el d, es la respuesta que resulta de su interpretación.

Estos ejemplos están relacionados con los que se presentan desde el constructor léxico.

1.- a) ¿LOS GATOS RESPIRAN POR LOS PULMONES?

b) p o1 r1 o2 p

c) p o1 r1 o2

d) CIERTO

2.- a) DIME QUE SON LOS PERROS

b) p r1 o1

c) p r1 o1

d) son MAMIFEROS

3.- a) ¿QUIENES COMEN CARNE?

b) q r1 o1 p

c) q r1 o1

d) LOS PERROS
LOS GATOS

4.- a) DIME ALGO SOBRE LOS DELFINES

b) p ol

c) p ol

d) son MAMIFEROS
respiran por LOS PULMONES
viven en EL MAR

5.- a) ¿QUE SABES SOBRE LOS PERROS?

b) p ol p

c) p ol

d) comen CARNE
son MAMIFEROS
respiran por LOS PULMONES

CONCLUSIONES

Se ha mostrado el desarrollo conceptual del manejo de conocimiento con el auxilio de la Inferencia Gramatical en donde el vocabulario y la interpretación de estructuras se aprende de lo que el usuario indica, los hechos se integran directamente en la RSA; en esta se puede aplicar la factorización para obtener un objeto genérico manteniendo las relaciones sintácticas en la red. La consulta se ejemplifica con una versión elemental de un Administrador.

Dado su carácter general algunas aplicaciones pueden ser:

- a) En la INVESTIGACION como base de hechos y auxiliar para obtener reglas para sistemas expertos
- b) En la EDUCACION como herramienta automatizada para consulta de información, interfase de sistemas automatizados de información bibliográfica y bancos de datos; como diccionario electrónico o apuntes inteligentes
- c) En la EMPRESA como sistema de consulta gerencial, consultor automatizado; como interfase para reservación de líneas aéreas, barcos, etc.
- d) En la INDUSTRIA para registro de información, auxiliar en obtención de Sistemas Expertos de control de procesos
- e) En el HOGAR como banco de datos familiar
- f) En los DEPORTES para el registro de datos históricos y obtención de sucesos semejantes.

Considerando las extensiones que puede tener, éstas pueden ser:

- a) Integración de otros mecanismos de percepción de la realidad: sonido, señales, etc.
- b) Adecuar el prototipo para que soporte de manera integrada los diferentes tipos de información
- c) Continuar el estudio lingüístico para dotar al sistema de un diálogo más natural.

ANEXOS

Esta parte del documento se presenta información que sirve de complemento al trabajo desarrollado. El anexo 1, describe brevemente algunos conceptos sobre gramáticas. El anexo 2, muestra la gramática con atributos completa que requiere el prototipo. El 3 y 4 muestran los listados de los programas usados en el modelo de agregación y del fractal del árbol.

ANEXO 1: GRAMATICAS

Se explican de manera sucinta algunos conceptos sobre gramáticas que se manejan en forma implícita en este documento.

Una GRAMATICA [12] es un sistema algebraico que describe el proceso mediante el cual se pueden generar oraciones de un lenguaje.

Consiste de cuatro elementos: un conjunto de símbolos no terminales o metavariabes, V_n ; un conjunto de símbolos terminales o alfabeto, V_t ; un conjunto de reglas o producciones, P ; y una metavariabie especial, S , llamada inicio o símbolo raíz, con ésta comienza la aplicación de las producciones.

Se representa generalmente por el cuádruple

$$G = \{ V_n, V_t, P, S \}$$

Ejemplo:

Sea la gramática $G = \{ V_n, V_t, P, O \}$ con

$$V_n = \{ O, S, P, A, \text{Sust}, V \}$$

$$V_t = \{ \text{el, la, un, una, gato, perro, cerca, come, salta} \}$$

$$P = \left\{ \begin{array}{ll} O & \rightarrow S \ P \\ S & \rightarrow A \ \text{Sust} \\ P & \rightarrow V \ S \\ A & \rightarrow \text{el} \mid \text{la} \mid \text{un} \mid \text{una} \\ \text{Sust} & \rightarrow \text{gato} \mid \text{perro} \mid \text{carne} \mid \text{cerca} \\ V & \rightarrow \text{come} \mid \text{salta} \end{array} \right\}$$

Algunas oraciones que se pueden derivar de ella son:

el gato come carne

el perro salta la cerca

Las gramáticas se agrupan de acuerdo con la jerarquía de Chomsky como se indica más abajo [15]. Las letras mayúsculas representan símbolos no terminales, las minúsculas símbolos terminales y las griegas cadenas de símbolos.

a) Restringidas o tipo cero.

Sus reglas son

$$\alpha \rightarrow \beta$$

donde $\alpha, \beta \in V^*$

$$V^* = \{ V_n \cup V_t \}$$

ejemplo:

$$\begin{array}{ll} ABC & \rightarrow xlm \\ abcd & \rightarrow edad \\ A & \rightarrow b \end{array}$$

b) Dependientes de contexto o tipo uno.

Sus reglas son

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

donde $\alpha, \beta, \gamma \in V^*$

$$A \in V_n$$

se dice que α y β le dan el contexto.

c) Libres de contexto o tipo dos, el lado izquierdo debe ser un sólo símbolo no terminal.

Sus reglas son

$$A \rightarrow \alpha$$

donde $A \in V_n$

$$\alpha \in V^*$$

d) Regulares o tipo tres, el lado izquierdo debe ser un sólo símbolo no terminal y el lado derecho un sólo terminal o un sólo terminal seguido por un sólo no terminal.

Sus reglas son

$$A \rightarrow aB \mid a \quad \text{ó} \quad A \rightarrow Ba \mid a$$

donde $A, B \in V_n$

$$a \in V_t$$

Las gramáticas de un tipo mayor pueden representarse mediante las de un tipo inferior, por ejemplo, un gramática de tipo dos puede derivarse de una de tipo uno haciendo el contexto vacío.

Las dos últimas se pueden reconocer con analizadores sintácticos (parsers) muy eficientes.

Dentro del estudio formal del lenguaje humano Chomsky propone tres tipos de reglas [18]:

- a) Reglas ahormacionales
- b) Reglas transformacionales
- c) Reglas morfofonémicas.

Las reglas ahormacionales son reglas como las que se encuentran en la definición de gramática al inicio de este anexo.

Las reglas morfofonémicas permiten que una cadena se exprese en término de fonemas.

Las reglas transformacionales se aplican sobre las estructuras que se obtienen con las reglas ahormacionales. Con estas reglas gramáticas ahormacionales que serían muy complejas se pueden plantear de una forma más sencilla.

Ejemplo:

Sea la gramática $G = \{ V_n, V_t, P, S \}$

$V_n = \{ S \}$

$V_t = \{ a, b \}$

$P = \{ \begin{array}{ll} S & \rightarrow aS \\ S & \rightarrow bS \\ \#xS\# & \rightarrow \#xx\# \end{array} \}$

aquí la tercera producción es una regla transformacional en la que el símbolo "x" representa una cadena.

Con esta gramática se pueden generar oraciones como:

abbabb

que resulta de la derivación:

$S \Rightarrow aS \Rightarrow abS \Rightarrow abbS \Rightarrow \#abbabb\#$

Una GRAMATICA CON ATRIBUTOS [13] es una gramática en la cual se intercalan en las producciones llamados a rutinas semánticas.

Se representa como:

$$GA = \{ V_n, V_t, V_s, P, S \}$$

donde

V_n , son variables no terminales;

V_t , son variables terminales;

V_s , son llamados a rutinas semánticas;

S , es la metavariante inicial;

P , son producciones con atributos de la forma

$$A \rightarrow \alpha$$

$$\alpha \in V^*$$

$$V^* = \{ V_n \cup V_t \cup V_s \}$$

ANEXO 2: GRAMATICA CON ATRIBUTOS DEL PROTOTIPO

Se muestra en este anexo la Gramática con Atributos completa que usa el Sistema Evolutivo para Representación del Conocimiento para interpretar las oraciones declarativas, interrogativas e imperativas.

La Gramática con Atributos es la siguiente:

$$GA = \{ V_n, V_t, V_s, P, S \}$$

donde

$$V_n = \{ A, B, C, O \}$$

$$V_t = \{ c, d, e, f, g, o, p, q, r, s, t, ;, . \}$$

$$V_s = \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$$

$$S = \{ A \}$$


```

P = {
  A -> O r O 1 B
  A -> p C
  A -> q r O 4
  A -> g 6
  A -> m 7
  A -> d 8
  A -> c 9
  A -> e 10
  B -> ; O r O 1 B
  B -> .
  C -> O r O 2
  C -> r O 3
  C -> O 5
  O -> o
  O -> f
  O -> t
  O -> s
}

```

En el seguimiento de esta gramática es importante hacerlo de arriba hacia abajo y de izquierda a derecha, porque es la manera en que se interpretan las oraciones del sistema. El no hacerlo de esta forma puede producir resultados erróneos.

Las rutinas semánticas resumidas son:

RUTINA	FUNCION	PARAMETROS		
		01	R	02
1	Integra hechos a la RSA	*	*	*
2	Indica si existe o no el hecho dado	*	*	*
3	Muestra los objetos apuntados por R desde 01 directa o indirectamente	*	*	/
4	Presenta los objetos que tengan asignados R 01 directa o indirectamente	*	*	/
5	Desplega las relaciones y objetos asociados directa o indirectamente al objeto 01	*	/	/
6	Activa el generalizador	/	/	/
7	Lista la RSA completa	/	/	/
8	Editor de imágenes	/	/	/
9	Conversión de imágenes	/	/	/
10	Editor de texto	/	/	/

El asterisco (*) indica que es un parámetro utilizado.

ANEXO 3: MODELO DE AGREGACION

En este anexo se describe el MODELO DE AGREGACION mediante el cual se puede simular el crecimiento de la RSA variando dos parámetros:

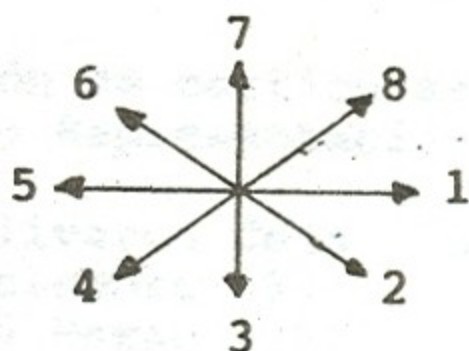
- a) El número de partículas
- b) el Factor de Densidad de Agregación(FDA).

Cada partícula representa un objeto de la RSA, los cuales llegan en forma aleatoria y se integran o quedan como objetos aislados también al azar.

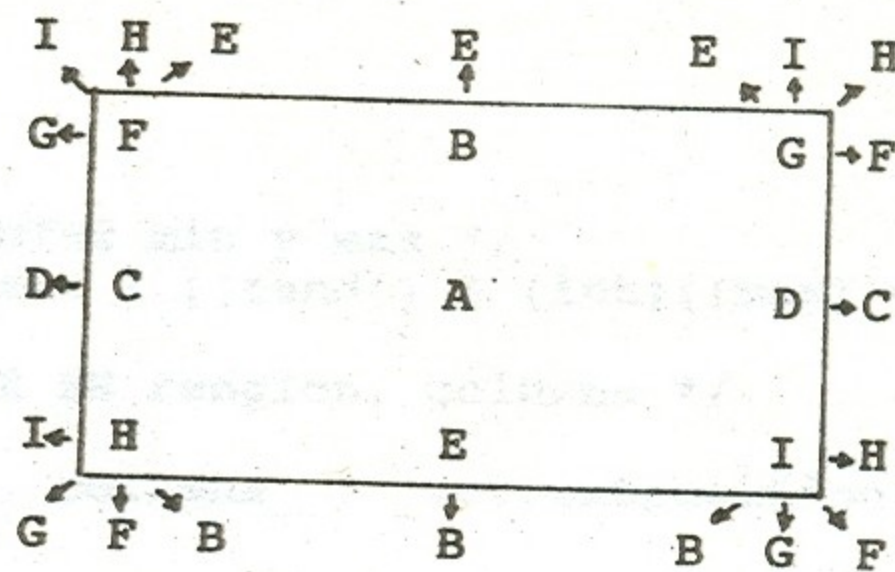
El FDA determina el nivel de relacionamiento entre los objetos; a medida que este número crece se considera que los objetos tienen mayor posibilidad de relacionamiento entre ellos.

En el modelo, la pantalla representa "la memoria" donde se almacena la RSA y cada pixel representa un objeto.

El desplazamiento de un pixel es en alguna de ocho direcciones posibles:



el desplazamiento se considera que es sobre una esfera, por lo que la pantalla se encuentra unida, según las posibilidades de movimiento de un pixel, de la siguiente forma:



El algoritmo del modelo es el siguiente:

```

Acepta parámetros
Genera la partícula inicial
Para cada partícula
{
  Genera su salida: (columna, renglón); distancia = 0
  Si tiene vecino
    se detiene la partícula
    siguiente partícula
  Si se debe detener por el FDA
    se detiene la partícula
    siguiente partícula
  Si distancia = 0
    distancia = RANDOM
    dirección = RANDOM
  Avanza pixel una posición según la dirección
  distancia = distancia - 1
}
Terminación del modelo
  
```

Este algoritmo se implementó en Quick C 2.0 y su listado se muestra a continuación:

```

/* MODELO2.c
   Modelo de Agregación de partículas.
   Sistema Evolutivo para Representación del Conocimiento
  
```

```

AUTOR: Jesús Manuel Olivares Ceja      Lenguaje C
FECHA DE CREACION: Diciembre 1990
ULTIMA MODIFICACION: 8 Marzo 1991
Versión 2.0
*/
  
```

```

#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <graph.h>
  
```



```

/*
DEFINICION DE MACROS
*/
/* NUMERO ALEATORIO ENTRE min y max */
#define RANDOM( min, max ) ((rand() % (int)((max) - (min))) + (min) + 1

/* POSICIONA EL CURSOR EN renglon, columna */
/*
#define PON( renglon_, columna_ ) _settextposition( renglon_, columna_
*/
/*
DEFINICION DE CONSTANTES
*/
#define SI 1
#define NO 0
#define COLMIN 10
#define COLMAX 309
#define RENMIN 3
#define RENMAX 181
#define RANGOREN (RENMAX - RENMIN)
#define RANGOCOL (COLMAX - COLMIN)
#define MAXINTER 9
#define MAXDIR 8
#define ACTIVO 3
#define LIBRE 0
#define DESPLAZAR 80
#define ERRONEO 10
#define SALIR_ 0x001B

/*
DEFINICION DE TIPOS DE DATOS
*/
typedef struct
{
    unsigned decol; /* COLUMNA INICIAL DEL INTERVALO */
    unsigned deren; /* RENGLON INICIAL DEL INTERVALO */
    unsigned hastacol; /* COLUMNA FINAL DEL INTERVALO */
    unsigned hastaren; /* RENGLON FINAL DEL INTERVALO */
} TIPO_INTR;

typedef struct
{
    unsigned col; /* VALOR A SUMAR A LA COLUMNA */
    unsigned ren; /* VALOR A SUMAR AL RENGLON */
} TIPO_POS;

/*
DECLARACION DE VARIABLES
*/
TIPO_INTR
    /* TABLA DE INTERVALOS */
    tinter[ MAXINTER ] =

```



```

{
  { COLMIN + 1, RENMIN + 1, COLMAX - 1, RENMAX - 1 }, /* A */
  { COLMIN + 1, RENMIN, COLMAX - 1, RENMIN }, /* B */
  { COLMIN, RENMIN + 1, COLMIN, RENMAX - 1 }, /* C */
  { COLMAX, RENMIN + 1, COLMAX, RENMAX - 1 }, /* D */
  { COLMIN + 1, RENMAX, COLMAX - 1, RENMAX }, /* E */
  { COLMIN, RENMIN, COLMIN, RENMIN }, /* F */
  { COLMAX, RENMIN, COLMAX, RENMIN }, /* G */
  { COLMIN, RENMAX, COLMIN, RENMAX }, /* H */
  { COLMAX, RENMAX, COLMAX, RENMAX }, /* I */
};

```

TIPO_POS

/* TABLA DE POSICIONAMIENTO */

tabpos[MAXINTER][MAXDIR] =

```

{
  /* A */
  { {1, 0 }, {1, 1 } },
    { {0, 1 }, {-1, 1 } },
    { {-1, 0 }, {-1, -1 } },
    { {0, -1 }, {1, -1 } }
  },
  /* B */
  { {1, 0 }, {1, 1 } },
    { {0, 1 }, {-1, 1 } },
    { {-1, 0 }, {-1, RANGOREN } },
    { {0, RANGOREN }, {1, RANGOREN } }
  },
  /* C */
  { {1, 0 }, {1, 1 } },
    { {0, 1 }, {RANGOCOL, 1 } },
    { {RANGOCOL, 0 }, {RANGOCOL, -1 } },
    { {0, -1 }, {1, -1 } }
  },
  /* D */
  { {-RANGOCOL, 0 }, {-RANGOCOL, 1 } },
    { {0, 1 }, {-1, 1 } },
    { {-1, 0 }, {-1, -1 } },
    { {0, -1 }, {-RANGOCOL, -1 } }
  },
  /* E */
  { {1, 0 }, {1, -RANGOREN } },
    { {0, -RANGOREN }, {-1, -RANGOREN } },
    { {-1, 0 }, {-1, -1 } },
    { {0, -1 }, {1, -1 } }
  },
  /* F */
  { {1, 0 }, {1, 1 } },
    { {0, 1 }, {RANGOCOL, 1 } },
    { {RANGOCOL, 0 }, {RANGOCOL, RANGOREN } },
    { {0, RANGOREN }, {1, RANGOREN } }
  },
};

```



```

/* G */
{ {-RANGOCOL, 0 }, {-RANGOCOL, 1 },
  {0, 1 }, {-1, 1 },
  {-1, 0 }, {-1, RANGOREN },
  {0, RANGOREN }, {-RANGOCOL, RANGOREN }
},
/* H */
{ {1, 0 }, {1, -RANGOREN},
  {0, -RANGOREN}, {RANGOCOL, -RANGOREN},
  {RANGOCOL, 0 }, {RANGOCOL, -1 },
  {0, -1 }, {1, -1 }
},
/* I */
{ {-RANGOCOL, 0 }, {-RANGOCOL, -RANGOREN},
  {0, -RANGOREN}, {-1, -RANGOREN},
  {-1, 0 }, {-1, -1 },
  {0, -1 }, {-RANGOCOL, -1 }
},
};

char salida[ 42 ];

/*
ACCION ERROR.- Se activa cuando se presenta algun error
*/
void accionerror( void )
{
    _setvideomode( _DEFAULTMODE );
    _outtext( "ERROR en MODELO DE AGREGACION" );
    exit(0);
}

/*
CUAL INTERVALO.- Regresa el intervalo que le corresponde a COLUMNA, REN
*/
unsigned cualinter( unsigned columna, unsigned renglon )
{
    unsigned inter;
    char area[ 40 ];

    for( inter = 0; inter < MAXINTER; inter++ )
        if( ((columna >= tinter[ inter ].decol &&
            columna <= tinter[ inter ].hastacol)) &&
            ((renglon >= tinter[ inter ].deren &&
            renglon <= tinter[ inter ].hastaren)) )
            return inter;
    return ERRONEO;
} /* CUAL INTERVALO */

/*
TIENE VECINO.- Determina si un pixel tiene algún vecino
*/
unsigned tienevecino( unsigned columna, unsigned renglon )
{
    unsigned inter, dir;

```



```

inter = cualinter( columna, renglón );
if( inter == ERRONEO )
    accionerror();
for( dir = 0; dir < MAXDIR; dir++ )
    if( _getpixel( columna + tabpos[ inter ][ dir ].col,
        renglon + tabpos[ inter ][ dir ].ren ) == ACTIVO )
        return SI;
return NO;
} /* TIENE VECINO */

/*
PON PIXEL.- Asigna el color ESTADO al pixel de RENglon, COLumna
*/
void ponpixel( unsigned col, unsigned ren, short estado )
{
    _setcolor( estado );
    _setpixel( col, ren );
} /* PON PIXEL */

/*
AVANZA PIXEL.- Desplaza una posición el pixel de COLumna, RENglon
*/
void avanzapixel( unsigned *col, unsigned *ren, unsigned dir )
{
    unsigned inter;

    inter = cualinter( *col, *ren );
    if( inter == ERRONEO )
        accionerror();
    ponpixel( *col, *ren, LIBRE );
    *col += tabpos[ inter ][ dir ].col;
    *ren += tabpos[ inter ][ dir ].ren;
    ponpixel( *col, *ren, ACTIVO );
} /* AVANZA PIXEL */

void main( void )
{
    unsigned particulas, particula, renp, colp, factor, direccion, distanci
    sigue, demora;

    /*
PRESENTACION DEL MODELO DE AGREGACION
*/
    _clearscreen( _GCLEARSCREEN );
    _displaycursor( _GCURSOROFF );
    PON( 6, 30 );
    _outtext( "MODELO DE AGREGACION" );
    PON( 8, 28 );
    _outtext( "Version 2.0 Febrero 1991" );
    PON( 20, 35 );
    _outtext( "JESUS MANUEL OLIVARES CEJA" );
    PON( 21, 35 );
    _outtext( "Av. Rio Mayo 21-A" );
    PON( 22, 35 );
    _outtext( "Fracc. Real del Moral" );

```



```

PON( 23, 35 );
_outtext( "09010, México, D. F. MEXICO" );
PON( 18, 1 );
_outtext( "Los comentarios y sugerencias favor de enviarlos a:" );
for( demora = 0; demora < 25000; demora++ )
;
/*
ACEPTA LOS PARAMETROS
*/
_displaycursor( _G_CURSORON );
_clearscreen( _G_CLEARSCREEN );
PON( 6, 35 );
_outtext( "PARAMETROS\n\n" );
printf( "MAXIMO DE PARTICULAS: " );
scanf( "%d", &particulas );
printf( "\n\nFACTOR DE DENSIDAD DE AGREGACION: " );
scanf( "%d", &factor );
_displaycursor( _G_CURSOROFF );
/*
INICIALIZACION DEL MODELO
*/
srand( (unsigned)time( NULL ) );
_setvideomode( _MRES4COLOR );
_setbkcolor( (long)LIBRE );
_clearscreen( _G_CLEARSCREEN );
PON( 25, 10 );
sprintf( salida, "PARTICULA _____ DE %d ", particulas );
_outtext( salida );
for( colp = COLMIN - 2; colp < (COLMAX + 2); colp++ )
{
    ponpixel( colp, RENMIN - 2, ACTIVO );
    ponpixel( colp, RENMAX + 2, ACTIVO );
}
for( renp = RENMIN - 2; renp < (RENMAX + 2); renp++ )
{
    ponpixel( COLMIN - 2, renp, ACTIVO );
    ponpixel( COLMAX + 2, renp, ACTIVO );
}
colp = RANDOM( COLMIN, COLMAX ) - 1;
renp = RANDOM( RENMIN, RENMAX ) - 1;
ponpixel( colp, renp, ACTIVO );
/*
ITERA DESDE LA SEGUNDA PARTICULA
*/
for( particula = 2; particula <= particulas; particula++ )
{
    PON( 25, 20 );

    sprintf( salida, "%5d", particula );
    _outtext( salida );
    do
    {
        colp = RANDOM( COLMIN, COLMAX ) - 1;
        renp = RANDOM( RENMIN, RENMAX ) - 1;
    }
}

```



```

while( _getpixel( renp, colp ) == ACTIVO );
distancia = 0;
ponpixel( colp, renp, ACTIVO );
for( sigue = SI; sigue; )
{
    if( teclasin() == SALIR_ )
    {
        _setvideomode( _DEFAULTMODE );
        exit(0);
    }
    if( tienevecino( colp, renp ) )
        sigue = NO;
    if( RANDOM( 0, factor ) == 1 )
        sigue = NO;
    if( sigue )
    {
        if( distancia < 1 )
        {
            direccion = RANDOM( 0, MAXDIR ) - 1;
            distancia = RANDOM( 0, DESPLAZAR );
        } /* if */
        if( teclasin() == SALIR_ )
        {
            _setvideomode( _DEFAULTMODE );
            exit(0);
        }
        avanzapixel( &colp, &renp, direccion );
        distancia--;
    } /* IF */
} /* for */
} /* FOR */
PON( 25, 1 );
sprintf( salida, "%d PARTICULAS, FDA = %d", partículas, factor );
_outtext( salida );
PON(25, 1 );
scanf( "%s", salida );
teclaon();
_settextcolor( _WHITE );
_setbkcolor( _BLACK );
_setvideomode( _DEFAULTMODE );
_displaycursor( _G_CURSORON );
_clearscreen( _G_CLEARSCREEN );
} /* MAIN */

```


Cuando ya se tiene la gramática generativa, al incluir otra oración canónica, se modifica de modo que genere las anteriores y la nueva.

El CONSTRUCTOR SEMANTICO asigna una interpretación a los elementos y oraciones del lenguaje.

Este mecanismo se ha implementado como sigue:

- a) Adicionando rutinas semanticas a la gramática generativa obteniendo una gramática con atributos [5] [25]
- b) Mecanismo de diálogo [1].- Es un proceso recursivo que cada vez que detecta un elemento desconocido en la oración pregunta por su significado en el mismo lenguaje que se utiliza

Usuario: CALCULA LA MEDIA DE 5, 3, 8 y 9

Constructor: Cómo se obtiene la 'MEDIA'

Usuario: SUMA LOS DATOS; LA SUMA LA DIVIDES ENTRE EL NUMERO DE DATOS

Constructor: ...

- c) Asociación de significado.- Se asocian diferentes mecanismos de captación de la realidad, por ejemplo borrador se asocia con la figura de un borrador
- d) Sinónimos.- Se asigna significado a un elemento en términos de otro (por ejemplo, GIS es sinónimo de TIZA).

El ADMINISTRADOR utiliza la imagen de la realidad obtenida por el constructor para dar solución a los problemas que se indiquen mediante un diálogo en lenguaje natural [2]. Este es el elemento deductivo [9], en su implantación se puede aprovechar alguna de las que se han desarrollado [13]: un Sistema Experto, una aplicación en C, etc.

ANEXO 4: FRACTAL DE UN ARBOL

Se muestra en este anexo el programa para obtener el fractal de un árbol.

El algoritmo recursivo que lo obtiene es como sigue:

Un árbol es un árbol a la izquierda, un árbol a la derecha y un árbol al centro.

Un árbol puede ser una rama.

El listado del programa implementado en el lenguaje Pascal versión 3.0.

```
{ ARBOL.pas
```

```
  Ejemplo de un objeto fractal (un Arbol).
```

```
  Sistema Evolutivo para Representación del Conocimiento
```

```
AUTOR: Jesús Manuel Olivares Ceja    Lenguaje PASCAL
```

```
FECHA DE CREACION: Diciembre 1990
```

```
ULTIMA MODIFICACION: 14 Febrero 1991
```

```
Versión 1.0
```

```
}
```

```
PROGRAM ARBOL(INPUT,OUTPUT);
```

```
CONST
```

```
  PI = 3.1416;
```

```
VAR
```

```
  N : INTEGER;
```

```
{
```

```
LINEA.- Traza una línea en modo gráfico
```

```
}
```

```
PROCEDURE LINEA(X, Y, XF, YF : REAL);
```

```
BEGIN
```

```
  DRAW(ROUND(X),ROUND(199-Y/2),ROUND(XF),ROUND(199-YF/2),1);
```

```
END; { LINEA }
```



```

{
ARBOL.- Dibuja un árbol de manera recursiva
}
PROCEDURE ARBOL( N, T : INTEGER; X, Y, A : REAL);
BEGIN
  IF N > 0 THEN
    BEGIN
      LINEA(X, Y, X+T/2*COS(A),Y+T/2*SIN(A));
      ARBOL
        (N-1, ROUND(T/2), X+T/2*COS(A), Y+T/2*SIN(A), A+(PI/4) );
      ARBOL
        (N-1, ROUND(T/2), X+T/2*COS(A), Y+T/2*SIN(A), A-(PI/4) );
      ARBOL(N-1, ROUND(T/2), X+T/2*COS(A), Y+T/2*SIN(A), A );
    END;
  END; { ARBOL }

{
SECCION PRINCIPAL
}
BEGIN
  HIRES;
  {
  ACEPTA EL NUMERO MAXIMO DE NIVELES DEL ARBOL MAYOR (3 a 7)
  }
  REPEAT
    GOTOXY( 1, 1);
    WRITE('ORDEN DEL ARBOL: ');
    GOTOXY( 18, 1 );
    READLN(N);
  UNTIL (N > 2) AND (N < 8);
  GOTOXY( 1, 1);
  WRITE(' ');
  LINEA( 0, 90, 639, 90 );
  {
  DIBUJA TRES ARBOLES
  }
  ARBOL(N, 300, 200, 90, PI/2);
  ARBOL(N-1, 150, 400, 90, PI/2 );
  ARBOL(N-2, 75, 520, 90, PI/2 );
  {
  TERMINACION DE LOS ARBOLES
  }
  GOTOXY( 17, 24 );
  WRITE( 'Figura 11.- Un ejemplo de objeto Fractal' );
  READLN;
  TextMode;
  ClrScr;
END.

```


BIBLIOGRAFIA

- [1] GALINDO Soria, Fernando, SISTEMAS EVOLUTIVOS DE LENGUAJES DE TRAYECTORIA en VI Reunión de Inteligencia Artificial, Memorias, Presidente José Negrete M., Ed. Limusa, Junio 1989, Querétaro, Qro.
- [2] GALINDO Soria, Fernando, SISTEMAS EVOLUTIVOS, IPN-UPIICSA Unidad de Investigación y Desarrollo, Junio 1985, México.
- [3] GALINDO Soria, Fernando, CONSTRUCCION DE SISTEMAS EVOLUTIVOS, IPN-UPIICSA.
- [4] GALINDO Soria, Fernando, CONFERENCIA TUTELAR, Introducción a los Sistemas Evolutivos, IPN-UPIICSA Departamento de Computación.
- [5] BERRUECOS Rodríguez, Elsa, SISTEMA EVOLUTIVO GENERADOR DE ESQUEMAS LOGICOS DE BASES DE DATOS, IPN-UPIICSA, México.
- [6] VICARIO Solórzano, Marina, AGUILAR Vallejo, Francisco Javier, EVA: EVOLUCION APLICADA, México.
- [7] AGUILAR Y AGUILAR, Araceli, Et. al, AVANCES Y EJEMPLOS DE SISTEMAS EVOLUTIVOS, IPN-UPIICSA Licenciatura en C. de la Informática, Julio 1988.
- [8] GALINDO Soria, Fernando, Notas del Seminario de Titulación "CONSTRUCCION DE SISTEMAS EVOLUTIVOS", IPN-UPIICSA, Octubre 1989 a Marzo 1990. México.
- [9] AGUILAR Y AGUILAR Araceli, Et. al, SISTEMAS EVOLUTIVOS, Conferencia en IPN-CENAC, 5 Agosto 1988. México.
- [10] ORTIZ Hernández, Javier, GALINDO Soria, Fernando, SISTEMAS EVOLUTIVOS CONSTRUCTORES DE SISTEMAS EXPERTOS, Centro de Investigación y Desarrollo Tecnológico, IPN-UPIICSA.
- [11] PEQUENO LAROUSSE ILUSTRADO 1987.
- [12] DICCIONARIO DE COMPUTACION.
- [13] GALINDO Soria, Fernando, PROGRAMACION DIRIGIDA POR SINTAXIS, IPN-UPIICSA-Lic. en C. de la Informática, Dic. 1985.
- [14] GALINDO Soria, Fernando, INFERENCIA GRAMATICAL (Hacia la matemática de la Informática), IPN-UPIICSA, México, D. F., DGIT-CENIDET, Cuernavaca, Mor., Oct. 1988.

- [15] NOTAS DEL CURSO DE LENGUAJES FORMALES, impartido por el Lic. Fernando Galindo Soria, tomadas por Jesús Manuel Olivares Ceja, IPN-UPIICSA, Sep. 1986-Feb 1987.
- [16] GALINDO Soria, Fernando, REPRESENTACION DEL CONOCIMIENTO, IPN-UPIICSA, Jul 1985, México.
- [17] RICH, Elaine, ARTIFICIAL INTELLIGENCE, Mc. Graw Hill, Singapoure, 1985.
- [18] CHOMSKY, Noam, ESTRUCTURAS SINTACTICAS, 9a. edición, introducción, notas, apéndice y traducción de C. P. Otero, Ed. Siglo XXI, México, 1987.
- [19] JULLIEN, Rémi, Botet, Robert y Kolb, Max, LOS AGREGADOS en Mundo Científico No. 75.
- [20] BOTET, Robert, Jullien, Rémi y Skjeltop, Arne T., LA FORMA RESULTADO DEL CRECIMIENTO en Mundo Científico No. 75.
- [21] SMITH & SMITH, DATA BASE ABSTRACTIONS OR AGREGATIONS AND GENERALIZATIONS, ACM TOOLS, Vol. 2.
- [22] SHANKAR, K. S., IBM, DATA STRUCTURES, TYPES AND ABSTRACCTIONS Apr. 1980 in Selected Reprints in Software, 2nd Edition, IEEE, 1985.
- [23] GALINDO Soria, Fernando, Comunicación personal sobre DISTRIBUCION DE CADENAS LINGUISTICAS, Diciembre 1990, IPN-UPIICSA, México.
- [24] El Increíble HULK en MICOMPUTER.
- [25] BERRUECOS Rodríguez, Elsa, SISTEMA EVOLUTIVO GENERADOR DE ESQUEMAS LOGICOS DE BASE DE DATOS, Seminario de Titulación, IPN-UPIICSA, México 1990.
- [26] GANASCIA, Jean-Gabriel, LA CONCEPCION DE LOS SISTEMAS EXPERTOS en Mundo Científico No. 53.
- [27] FAGOT Largeault, Anne, LA SIMULACION DEL RAZONAMIENTO MEDICO en Mundo Científico No. 53.
- [28] LOPEZ DE MANTARAS Badia, Ramón, MODELOS DE RAZONAMIENTO APROXIMADO en Mundo Científico No. 53.
- [29] LESORT, Marc, COMO DESCUBRI LOS FRACTALES entrevista a Benoît Mandelbrot en Mundo Científico No.58.
- [30] GALLAIRE, Herve, LA REPRESENTATION DES CONNAISSANCES en La Recherche No. 170, Octubre 1985.

