3. ECODE

El sistema Ecode fue diseñado con la intención de extraer contextos definitorios a partir de textos de especialidad procesados y seleccionados.

Alarcón (2009) desarrolló un sistema que procesa textos y entrega un conjunto de contextos definitorios, mismos que son identificados en un texto etiquetado con las categorías gramaticales (POS) de las palabras. Así, Ecode entrega una lista de CDs con etiquetas estilo XML, mismos que son validados cualitativamente en un archivo de texto plano (Sierra y Alarcón, 2010).

En este capítulo se analiza y describe el sistema de Alarcón (2009), se destaca su funcionamiento, se describen los errores y finalmente se proponen ajustes de optimización y adaptación según las necesidades de los usuarios.

3.1. Descripción y análisis de Ecode

Ecode es un sistema desarrollado en Perl que, a grandes rasgos, utiliza gramáticas y expresiones regulares para extraer patrones lingüísticos en un archivo de texto etiquetado con POS, además de que proporciona una lista de CDs etiquetados en un archivo de texto.

Ecode se compone de nueve módulos, tres gramáticas y su respectivo script de ejecución. Estos módulos son los siguientes:

Módulo	Descripción
01_vds.pm	Etiquetado de verbos definitorios en candidatos
02_pvds.pm	Marcado de patrones verbales definitorios (PVDs)
03_td.pm	Tipo de definición de los candidatos
04_filtro.pm	Filtrado de contextos no relevantes
05_arbol.pm	Árbol de decisión para la identificación de término y definición
06_retagging.pm	Limpieza de etiquetas
07_rankingTyD.pm	Evaluación individual de los términos y definiciones
08_rankingGlobal	Evaluación global de los contextos
09_final.pm	Última limpieza y presentación de resultados

Tabla 3.1 Módulos del Ecode

Por otra parte, las gramáticas son:

Gramática	Descripción
01_gramaticaPOS.pm	Gramática de partes de la oración (módulo de Perl).
02_gramPVDs.txt	Gramática de patrones verbales definitorios y sus diferentes tipos y posiciones.
03_gramFiltro.txt	Gramática de reglas de filtrado de contextos no relevantes.

Tabla 3.2 Gramáticas de Ecode

El proceso de identificación de contextos definitorios se conforma de 3 partes que son:

- Identificación de candidatos (etiquetado de verbos definitorios, PVDs y tipo de definición)
- Filtrado e identificación de término y definición
- Evaluación y presentación de resultados

El diagrama del sistema se puede resumir en la siguiente figura:

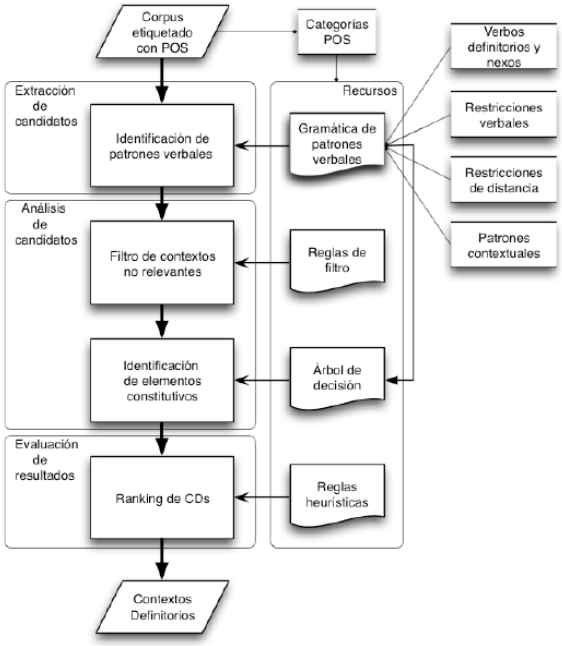

Figura 3.1 Panorama general de la arquitectura de Ecode (Alarcón 2009, 143)

En este punto vale la pena aclarar que el sistema no posee una interfaz de usuario, ni un menú con opciones en consola, sino que simplemente se ejecuta desde la terminal con un archivo de argumento y entrega un segundo archivo de texto con los CDs de salida.

3.1.1. Entrada al sistema

Para iniciar la extracción, el sistema requiere de un archivo de texto separado por líneas en las que cada línea es una oración; una oración se entiende para el fin del sistema como un fragmento textual entre signos de puntuación (como punto, punto y coma o un salto de línea) obtenida de un texto. Cada línea requiere tener una etiqueta que marca el inicio de línea y puede contener información relevante al origen de esta o una numeración.

Por ejemplo:<*doc_codi 00000*>*: La/6 banda/8 de/4 ADN/B que/0 es/9 transcrita/8 se/5 denomina/9 banda/8 codificante/8 ./s* En donde <doc_codi 00000> es una etiqueta de inicio de línea y "La/6" es una palabra etiquetada con su categoría gramatical.

Para el etiquetado de las palabras con su categoría gramatical (POS) se han utilizado diferentes etiquetadores; sin embargo, el conjunto de categorías gramaticales que requiere Ecode no cambia y está basada en el conjunto de etiquetas POS del Corpus del Español Mexicano Contemporáneo (CEMC), aunque originalmente fue pensado para trabajar con el estándar de etiquetado EAGLES. Esto que permite adaptar otros etiquetadores POS al

sistema con una simple sustitución de sus etiquetas. Al iniciar el presente trabajo, el etiquetador POS utilizado era el del CEMC desarrollado por Luis Fernando Lara, basado en el algoritmo de Brill. Veamos las posibles categorías gramaticales que genera este etiquetador:

Codificación numérica	**Categoría**	**Codificación abreviada**
0	Ambigua	Amb
1	Adverbio	Adv
2	Adjetivo	Adj
3	Conjunción	Con
4	Preposición	Pre
5	Pronombre	Pro
6	Artículo	Art
7	Contracción	Ctr
8	Nominal	Nom
9	Verbo	Vbo
A	Apoyos conversacionales	Apc
B	Nombres propios	Npr
C	Otros, como cifras, errores y palabras que comenzaban con mayúscula	Otr

Tabla 2.1[10] Conjunto de marcas del CEMC (Méndez, 2009: 75)

El etiquetador POS, no forma parte del Ecode, por lo que es un proceso externo previo a la ejecución que, para funcionar, requiere las siguientes características:

- La entrada debe tener los signos de puntuación separados de las palabras para su correcto etiquetado. Por ejemplo, "La dopamina no es una proteína; sí lo son, en cambio, las enzimas que sintetizan este neurotransmisor" se debe cambiar por "La dopamina no es una proteína; sí lo son , en cambio , las enzimas que sintetizan este neurotransmisor ."
- Todas las palabras deben estar en minúsculas, de lo contrario pueden ser etiquetadas de manera incorrecta.
- La entrada al etiquetador POS debe codificarse en caracteres iso-8859-1 y es muy

[10] Esta tabla proviene del capítulo 2

sensible a los errores de codificación ocasionando, en gran medida, por un etiquetado incorrecto.

3.1.1.1. Problemas y errores detectados

Los problemas y errores detectados en el proceso son los siguientes:

1. El etiquetador POS es lento, ineficiente e imprime una gran cantidad de texto mientras procesa. Genera archivos intermedios en vez de trabajar en memoria. Además no está integrado de ninguna manera a Ecode.
2. La entrada que recibe es muy específica por lo que los usuarios encuentran dificultad de alimentar el sistema, además de que no tiene un menú, ni interfaz, y sólo recibe texto en bruto, previamente etiquetado con POS, por lo que no es flexible ni accesible. La mayoría del texto se encuentra en archivos de aplicación.
3. ¿Y el punto tres? ¿Se queda vacío?

3.1.1.2. Líneas de trabajo propuestas

Las líneas de trabajo que se propusieron son las siguientes:

1. Desarrollar un módulo de obtención de texto que permita extraer texto en bruto de diferentes fuentes comunes como archivos de aplicación (PDFs, documentos de MSWord, XMLs, HTMLs), de direcciones directamente de Internet (URLs), bases de datos de otros desarrollos (Describe®, Corpus de las Sexualidades en México) y archivos de texto en bruto (txt).
2. Implementar la recepción del texto en bruto e identificar su codificación de caracteres para convertirlo en una codificación adecuada y universal como salida del Ecode, lo más adecuado sería: UTF-8.
3. Crear un separador de oraciones y filtrador de texto basura para evitar la carga del proceso principal y desechar oraciones muy grandes, muy pequeñas o mal estructuradas.
4. Implementar, ya sea un algoritmo o un programa, que etiquete con POS y sea más eficiente, rápido y con mayor precisión en su etiquetado, que trabaje completamente en la memoria y esté plenamente acoplado al sistema.
5. Entrenar y probar dicho etiquetador POS con un corpus de prueba consistente en la posible entrada que los usuarios emplearían para maximizar la precisión del mismo.

3.1.2. Algoritmo General

Al recibir el texto de entrada previamente etiquetado con POS el sistema realiza un proceso principal que consiste en los procesos descritos en la tabla 3.1 y se compone de tres módulos que son:

1. etiquetado de verbos y patrones verbales definitorios y la determinación del tipo de definición
2. filtrado e identificación de los términos y sus definiciones
3. la evaluación y presentación de resultados

A continuación se describe el algoritmo de dichos módulos.

3.1.2.1. Etiquetado de verbos definitorios, patrones verbales definitorios y tipo de definición

(Sierra y Alarcón, 2010) El primer proceso dentro del Ecode es la extracción de candidatos a contextos definitorios:

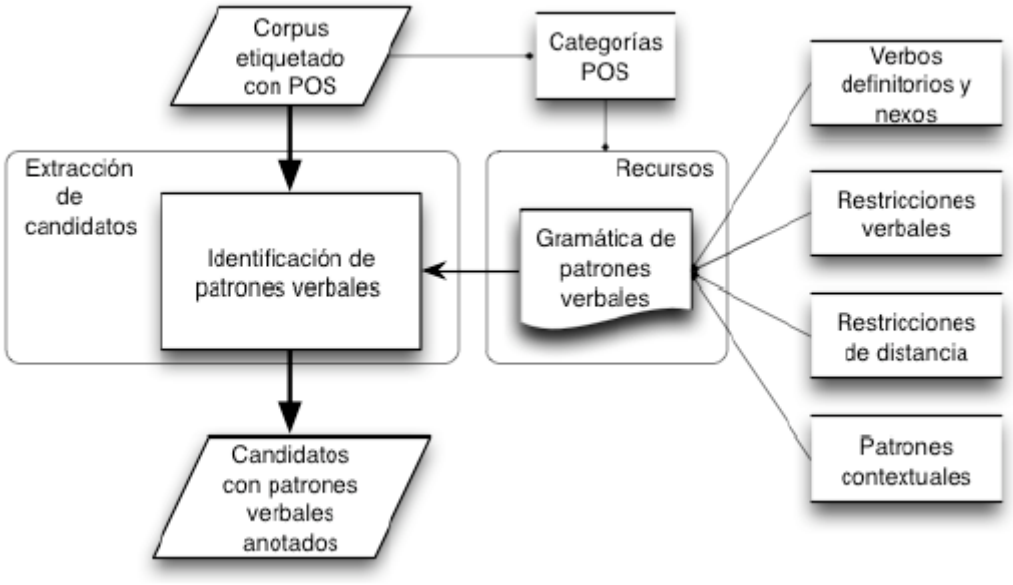

Figura 3.2 Extracción de candidatos a CD (Alarcón 2009: 146)

La entrada es, como ya se mencionó, un archivo de texto con “oraciones” etiquetadas con POS y una etiqueta de inicio de línea. Para la extracción de candidatos se utilizan 2 “gramáticas”: la primera, la gramática de categorías POS (Archivo: 01_gramaticaPOS.pm) que es un módulo de Perl con la definición de las categorías POS y algunos patrones que serán usados durante todo el programa; y la segunda, la gramática de patrones verbales (Archivo: 02_gramPVD.txt) que es un archivo de texto en donde se definen los patrones verbales y las características de ellos, a saber: verbos definitorios que los producen, restricciones a los tiempos verbales, restricciones de distancia entre palabras y patrones conceptuales.

La gramática de categorías POS si bien define reglas de cómo deben estar etiquetadas las palabras, contiene las categorías gramaticales relevantes para el análisis de candidatos y declara variables para uso global en el sistema, es decir, no es una gramática propiamente dicha. En primer lugar, el código de esta contiene estructuras de control y programación orientada a manipular el texto de entrada y no las definiciones de estructuras gramaticales requeridas. Además contiene definiciones de variables que no son categorías POS, pero que sí se requieren a lo largo del programa (por ejemplo, patrones pragmáticos) y también es aprovechada para inicializar arreglos globales.

La gramática de patrones verbales es un archivo de texto con estructura similar a XML que contiene definiciones de los posibles verbos con sus tiempos verbales y demás parámetros para cada tipo de definición de CD. Los parámetros que contiene son:

Parámetro	Posibles Valores	Descripción
Td	Ana, ext, fun, sin	Tipo de Definición
Lm	El infinitivo del verbo	Lema del Verbo Definitorio
Raíz	Regex de la raíz del verbo	Raíz de Verbo Definitorio
raízEx	Regex de la raíz del verbo	Excepción de la Raíz del Verbo Definitorio
Dist	0..n, any	Distancia entre el Verbo Definitorio y su Nexo
Nx	0 | como, por en, etc. ...	Nexo del Patrón Verbal Definitorio
Lt	I=izquierda; N=Nexo; D=Derecha	Lugar del Término dentro del Patrón Verbal Definitorio
lnx	Any	Lugar del Nexo respecto al Verbo Definitorio

Tabla 3.3 Parámetros de la gramática de patrones verbales.

El sistema utiliza estas gramáticas para encontrar los verbos definitorios y este proceso se efectúa en tres pasos, cada uno programado en un módulo de Perl (Archivos: 01_vds.pm, 02_pvds.pm y 03_td.pm):

1. Etiquetado de verbos definitorios (Archivo: 01_vds.pm)

El primer paso es etiquetar las excepciones de verbos definitorios definidos en la gramática de patrones verbales:

Para cada línea en el archivo de entrada

Si la línea contiene una excepción

Se etiqueta la excepción con: <vdEX>excepción</vdEX>

Después el sistema etiqueta los verbos definitorios a partir de las expresiones regulares de las raíces definidas en la gramática:

Para cada línea en el archivo de trabajo

Si la línea empareja con una regex de raíz verbal

Etiqueta el verbo con <vd lemma="lema del verbo">verbo</vd>

2. Etiquetado de los patrones verbales definitorios (PVDs) (Archivo: 02_pvds.pm)

Como ya que se han etiquetado los verbos definitorios, el siguiente paso es etiquetar todos los elementos constitutivos de los patrones, incluyendo los verbos auxiliares, los pronombres, los nexos y las posibles palabras entre el verbo y su nexo:

Para cada patrón verbal definitorio en la gramática

Por cada línea en el arreglo de trabajo que contiene un verbo etiquetado

Si un patrón verbal es encontrado

Etiqueta el nexo con <nx>nexo</nx>

Etiqueta los verbos auxiliares con <aux>verbo auxiliar</aux>

Etiqueta los pronombres con <pr>pronombre</pr>

Etiqueta todo el patrón entre <pdv>patrón verbal</pdv>

3. Etiquetado del tipo de definición, este proceso consiste en etiquetar el tipo de definición de cada candidato considerando la información de la gramática de patrones verbales:

Por cada patrón verbal definitorio en la gramática

Para cada línea del arreglo de trabajo que contiene un patrón etiquetado

Si el lema anotado coincide con el tipo de definición especificado en la gramática

Etiqueta el candidato con <tipoD=”tipo de definición”/>

Un ejemplo de un candidato adecuadamente anotado es:

<tipoD= “analitica”/> El metabolismo<pvd> <pr>se</pr> <aux>puede</aux> <vd lemma= “definir”> definir</vd> <nx>como</nx></pvd> la suma de todos los procesos químicos (y físicos) implicados.

3.1.3. Filtrado e Identificación de término y definición

(Sierra y Alarcón, 2010)Para preparar el análisis de los candidatos se marcan con etiquetas contextuales que sirven como bordes para los procesos automáticos que se llevan a cabo. Todo a la izquierda del patrón verbal es anotado entre <izq /> y todo a la derecha con <der />. Ejemplo:

<tipoD=“analitica”/> <izq>El metabolismo</izq> <pvd><pr>se</pr> <aux>puede</aux> <vd lemma=“definir”>definir</vd> <nx>como</nx><dvp> <der>la suma de todos los procesos químicos (y físicos) implicados.</der>

Después de dicho etiquetado el siguiente paso se divide en 2 procesos: uno, el filtrado de candidatos no relevantes; y el otro, la identificación del término y definición en los CDs.

En el primer caso se aplica un filtro tomando en cuenta que los patrones definitorios pueden ser usados en un rango más amplio de oraciones. En el caso de patrones verbales, algunos verbos tienden a tener un significado metalingüístico más amplio que otros. En el caso de *definir* o *denominar* vs. c*oncebir* o *identificar*, los dos últimos son usados en una variedad de contextos que no necesariamente están expresando información léxica sino…? Además de que algunos verbos con un significado metalingüístico más alto no sólo son usados para definir términos.

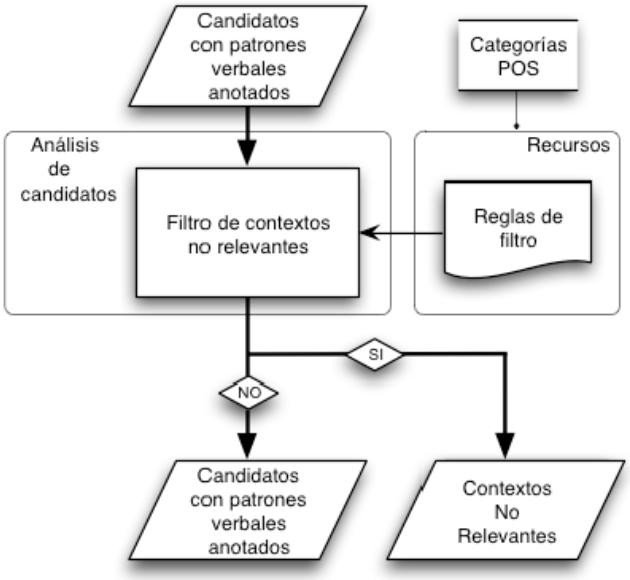

Figura 3.3 Filtro de candidatos no relevantes (Alarcón, 2009: 165)

En este proceso, la entrada consiste en candidatos a CDs con patrones verbales etiquetados. El principal recurso usado para filtrar candidatos no relevantes es un conjunto de reglas de filtrado que usan algunas categorías gramaticales. Cuando los candidatos validan ante alguna de estas reglas, son considerados como candidatos no relevantes. Ejemplos.

Existen partículas o secuencias sintácticas que pueden aparecer cuando los patrones verbales definitorios no son usados para definir un término. Estas partículas y secuencias fueron encontradas en posiciones especificas, por ejemplo: algunas partículas de negación como “no” o “tampoco” que se encontraron en la primera posición antes o después del PVD; los adverbios como “tan”, “poco”, así como secuencias como “poco más”, fueron encontrados entre el verbo definitorio y el nexo “como”, además, se encontraron estructuras o secuencias sintácticas de adjetivo + verbo en la primera posición, después del verbo definitorio.

Posición	Regla
Izquierda	para </izquierda>
Nexo	</dv> .* verbo conjugado .* </nx>
	</dv>.*? se .*?<nx>
	</dv>.*? tanto .*?<nx>
	</dv>.*? sino .*?<nx>
	</dv> , <nx>
	así <nx>
	cerca <nx>
	parte <nx>
	partir <nx>
	más <nx>
	menos <nx>
	mientras (que ,|que) <nx>
	no <nx>
	poco <nx>
	poco más <nx>
	(que ,|que) <nx>
	sino <nx>
	tales <nx>
	tal <nx>
	y <nx>
	ya <nx>
Derecha	<derecha> antes
	<derecha> cuan
	<derecha> para
	<derecha> si
	<derecha> se
	<derecha> verbo conjugado

Tabla 3.4 Reglas de filtrado de contextos no relevantes (Alarcón 2009:167).

El sistema utiliza estos parámetros para filtrar contextos no relevantes. Este proceso se realiza en dos pasos:

1. Etiquetando verbos definitorios. En el que el algoritmo se apareja con todas las instancias de reglas de filtrado encontradas en los candidatos.

Para cada candidato con patrones definitorios verbales
Para cada secuencia en las reglas de filtrado
Si el candidato se apareja con la secuencia
Etiquetar la secuencia con <filtro>secuencia</filtro>

2. Excluir excepciones de la lista de candidatos. Sólo los candidatos no filtrados se dejan después de este paso.

Para cada candidato con patrones definitorios verbales
Si el candidato contiene un filtro
Excluir el candidato de la lista

Un ejemplo de un contexto filtrado como no relevante es el siguiente:

Regla: para </Izq>

<izq>Para</izq><pvd><vd lemma="entender">entender</vd> <nx>como</nx></pvd> <der> funciona el ADN, es necesario conocer algo sobre su estructura y organización</der>.

Una vez que los contextos no relevantes se han filtrado, el siguiente proceso del análisis de candidatos, es la identificación del término y la definición.

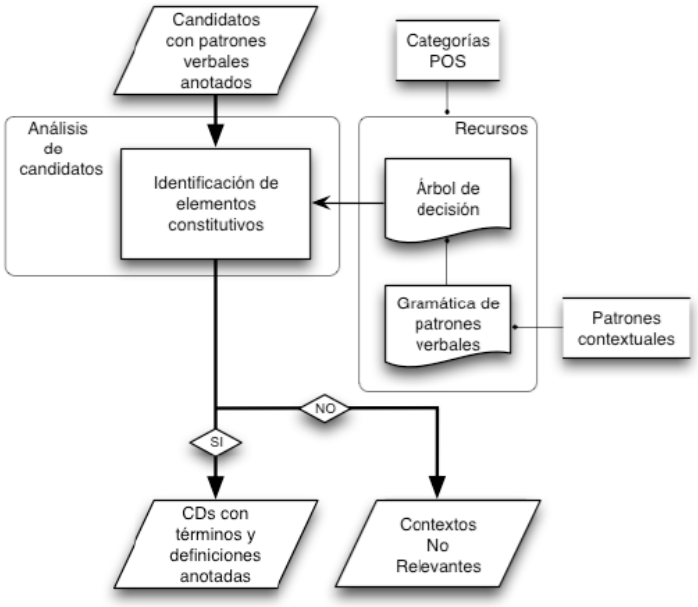

Figura 3.4 Identificación de términos y definiciones (Alarcón 2009: 170)

En este proceso, la entrada consiste en los candidatos a CDs restantes que no han sido filtrados como contextos no relevantes. La identificación del término y la definición se realiza con un árbol de decisión, que toma como fuente principal los patrones contextuales en la gramática de patrones verbales. La entrada también está condicionada ¿como en? el proceso previo, es decir, si los candidatos se aparejan con una de las reglas del árbol de decisión, se consideran como CDs válidos; si no, se filtran como contextos no relevantes. La salida en este caso consiste en CDs con términos y definiciones localizadas.

El uso de patrones contextuales en esta etapa se realiza considerando que, dependiendo de cada PVD, los términos pueden aparecer en algunas posiciones específicas en CDs en español. Algunos verbos definitorios permiten diferentes configuraciones estructurales; entonces, el proceso de identificarlos automáticamente y a sus definiciones está altamente relacionado con la posición de cada elemento constitutivo.

Se utiliza un árbol de decisión para resolver este problema, para detectar por medio de inferencias lógicas las posiciones probables de los términos, definiciones y patrones pragmáticos. Para asimilar esto, se establecieron algunas expresiones regulares simples para representar cada elemento constitutivo:

Elemento	Expresión regular	Descripción
ERT	TRE = BRD (Det) + N + Adj. {0,2} .* BRD	Expreg de Término
ERPP	PPRE = BRD (sign) (Prep | Adv) .* (signo) BRD	Expreg de Patrón Pragmático
ERD	DRE = BRD (Det) + N	Expreg de definición
N	Se obtiene de las etiquetas POS	Nombre
Adj	Se obtiene de las etiquetas POS	Adjetivo
Prep	Se obtiene de las etiquetas POS	Preposición
Adv	Se obtiene de las etiquetas POS	Adverbio
BRD	Es izq o der	Borde

Tabla 3.5 Elementos de la identificación de términos y definiciones

Como en el proceso de filtrado, las etiquetas contextuales funcionan como límites para marcar las instrucciones del árbol de decisión. Además, cada expresión regular podría funcionar como un límite. Una representación de un árbol de decisión se puede apreciar en la siguiente figura:

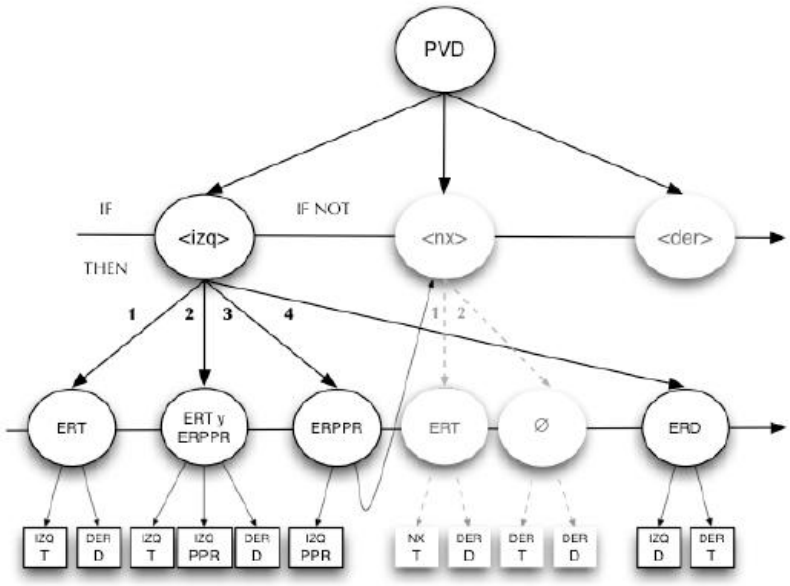

Figura 3.5 Árbol de decisión para la identificación de términos y definiciones (Alarcón, 2009: 175)

En el primer nivel, las ramas del árbol corresponden a posiciones diferentes en las cuales pueden ocurrir (nexo, derecha o izquierda). En el segundo nivel, las ramas corresponden a las expresiones regulares de cada elemento del CD. Los nodos (conjunciones de ramas) corresponden a decisiones tomadas de los atributos de cada rama y también se relacionan horizontalmente con inferencias *SI* o *SI NO*, y verticalmente a través de inferencias *ENTONCES.* Finalmente, las hojas corresponden a la posición asignada de cada elemento constitutivo. Las inferencias del árbol de decisión se explican en la siguiente tabla:

Posición	**Si**		**Entonces**
Nexo		La posición de nexo corresponde solamente a una expresión regular de término:	<nx> = término; <derecha> = definición
		La posición de nexo corresponde a un término y a una expresión regular de patrón pragmático:	<nx> = término; <nx> = patrón pragmático; <derecha> = definición
		La posición de nexo corresponde solamente a una expresión regular de patrón pragmático:	Ir a inferencia 4
Derecha		La posición derecha corresponde a un término y a una expresión regular de definición	<derecha> = término; <derecha> = definición
		La posición derecha corresponde solamente a una expresión regular de término	<derecha> = término; <izquierda> = definición
		La posición derecha corresponde solamente a una expresión regular de definición	Ir a la inferencia 7
Izquierda		La posición izquierda corresponde a un término y a una expresión regular de patrón pragmático.	<izquierda> = término <izquierda> = patrón pragmático; <derecha> = definición
		La posición izquierda corresponde solamente a una expresión regular de término	<izquierda> = término; <derecha> = definición

Tabla 3.6 Inferencias del árbol de decisión (Sierra y Alarcón, 2010: 10)

Para ejemplificar el proceso previo se puede observar los siguientes dos contextos Ejemplo:

> *<izquierda>En sus comienzos</izquierda> <dvp>se <dv lemma="definir"> definió</dv></dvp> <nx>la psicología como</nx> <derecha>"la descripción y la explicación de los estados de conciencia" (Ladd, 1887).</derecha>*

Esto es, el árbol infiere la regla 1: la posición del nexo corresponde solamente a una expresión regular de término. Entonces la posición del nexo corresponde al término (*la psicología*) y la posición derecha corresponde a la definición (*la descripción y la explicación de los estados de conciencia*….). Ejemplo:

> *<izquierda>Una librería genómica</izquierda> <dvp>se <dv lemma="definir"> define</dv></dvp> <nx>tradicionalmente como</nx> <derecha>un conjunto de clones en el que está representado todo el genoma de un organismo.</derecha>*

Aquí el árbol analiza las inferencias 1 a 3 y encuentra la regla 3: la posición de nexo corresponde solamente a una expresión regular de patrón pragmático (*tradicionalmente*); entonces analiza las inferencias 4 a 8 y encuentra la regla 8: la posición izquierda corresponde solamente a una expresión de término regular (*una librería genómica*) y la posición derecha corresponde a la definición (*un conjunto de clones en el que*…).

3.1.3.1. Ranking y presentación de resultados

La evaluación y presentación de resultados se conforma de 4 módulos (Archivos: 06_retagging.pm, 07_rankingTyD.pm y 08_rankingGlobal.pm, 09_final.pm).El sistema incluye un evaluador automático de resultados de los candidatos que fueron identificados como CDs. Este evaluador identifica los contextos con estructuras más prototípicas de términos y definiciones.

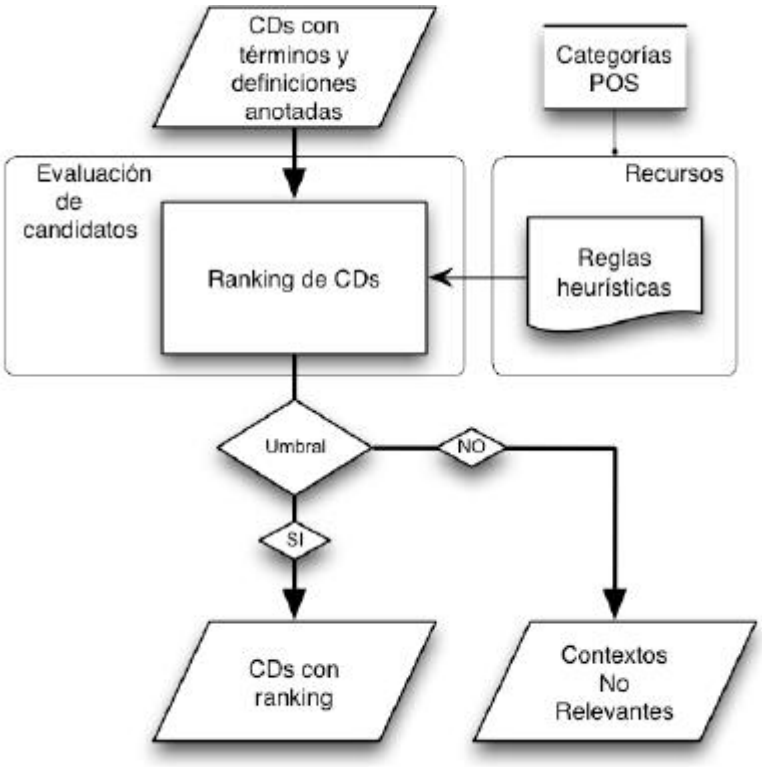

Figura 3.6 Evaluación de CDs (Alarcón, 2009:188)

Donde la entrada consiste en candidatos que fueron clasificados por el sistema como CDs. Para realizar la evaluación, el algoritmo usa algunas categorías POS para conformar un conjunto de reglas heurísticas que analizan la estructura sintáctica de los elementos automáticamente clasificados como términos o definiciones. Además de lo anterior, el algoritmo asigna valores numéricos para cada elemento constitutivo y estos valores se combinan en un elemento general que determina si cada CD pasa o no de acuerdo a un umbral preestablecido.

La salida final se condiciona: si el CD pasa el umbral, entonces se considera como un CD válido; si no, se considera como un contexto no relevante.

Las reglas heurísticas del procedimiento de evaluación se aprecian en la siguiente tabla:

Valor	Regla
Reglas de Término	
1	<t>$comillas.*$comillas</t>
1	<t>.*$parentesis $termino $parentesis</t>
2	<t>$coma.*$coma .*</t>
2	<t>$parentesis.*$parentesis .*</t>
3	<t>$demos\|$demos.*</t>
3	<t>.*$pron.*</t>
Reglas de definición	
1	<d>.* $que $verboConjugado</d>
1	<d2>$verboConjugado.*</d2>
2	<d>$palabra {,5}</d>
2	<d>.* (;|$noObstante| $sinEmbargo) .*</d>
3	<d>$demos</d>
3	<(d|d1)>NULL</(d|d1)>

Tabla 3.7 Reglas de evaluación de término y definición (Alarcón 2009: 189-191)

Se pueden observar diferentes reglas que asignan diversos valores a la estructura de términos y definiciones en la tabla 3.8. El valor 1 significa el mejor resultado, mientras que el 3 significa el peor; los candidatos que no siguen ninguna de las reglas son asignados con el valor 2. Por ejemplo, en el caso de estructuras de términos, el valor 1 es asignado a los casos que están marcados entre comillas, mientras que el valor 3 se asigna a aquellos candidatos cuya estructura de término consiste en un pronombre que pudiera indicar una posible referencia anafórica. En el caso de reglas de definición, el valor 1 es asignado a estructuras donde una clausula relativa se introduce después del pronombre "que", que puede ser una estructura prototípica en definiciones analíticas, mientras que un valor de 3 se asigna a los casos que solamente consisten en un pronombre demostrativo.

El algoritmo utiliza estas reglas para evaluar a cada candidato. Este proceso se realiza en tres pasos:

1. Evaluación de secuencias de términos y definiciones

    Para cada secuencia de término y definición en los candidatos

        Para cada regla de término y definición

            Si la secuencia de término y definición combinan con la regla

Etiquetar la secuencia con su valor correspondiente

2. Evaluar cada candidato con un valor global

Después de que todos los candidatos tienen su término y definición evaluados, el sistema asigna un valor global siguiendo las siguientes reglas:

Valor de evaluación de término y definición	Valor global de evaluación
<t = 1> y <d = 1>	<rG = 1>
<t = 2> o <d = 2>	<rG = 2>
<t = 2> y <d = 2>	<rG = 3>
<t = 3> o <d = 3>	<rG = 4>

Tabla 3.8 Reglas globales de evaluación de CDs

3. Excluir candidatos que no sobrepasan el umbral definido.

En este caso, he establecido el umbral en el valor 4.

Algunos ejemplos de los elementos de términos y definiciones evaluados son los siguientes:

Valor	**Ejemplo**
	Reglas de Términos
	<t v="1">«intrones»</t>
	<t v="3" >Este cloroplasto</t>
	Reglas de definición
	<t>la mutación rutabaga</t> <dvp>es </dvp> <d v="1">una mutación errónea que destruye a la adenilciclasa, interrumpiendo la síntesis del AMPc</d>.
	<d v="3">Esto</d> <dvp>se conoce <nx>como</nx></dvp> <t>mutación</t>.

Tabla 3.9 Ejemplo de términos y de definiciones evaluados (Alarcón 2009: 190,191)

De la tabla 3.10. Se puede observar que el ejemplo del término evaluado con 1 es valorado como un buen candidato por los marcadores tipográficos que enfatizan su presencia como un elemento importante en el texto; mientras que el ejemplo del término con valor de 3 es evaluado como el peor candidato porque el pronombre "este" se usa para hacer una referencia a un elemento que ha sido presentado anteriormente. Por otro lado, el primer ejemplo de las definiciones evaluadas se valora con un puntaje alto ya que tiene la

presencia del pronombre “que”, que está introduciendo la *differentia* en una definición analítica; finalmente, el segundo ejemplo se considera con valor 3 porque su estructura sintáctica que fue clasificada como una definición corresponde solamente a un pronombre relativo que está introduciendo una referencia anafórica.

Como salida se produce una lista de contextos definitorios etiquetados con sus partes distintivas en orden de calificación de su evaluación; esto consiste solamente en texto plano con etiquetas parecidas a XML. Un ejemplo de contexto etiquetado sería el siguiente:

<cd rG="1"><t>el chahuistle</t> <pvd><vd lema="ser">es</vd></pvd> <d>una enfermedad del maíz , conocida a la perfección por las comunidades prehispánicas que se dedicaban a la siembra de esta planta.</d></cd>

3.1.3.2. Problemas detectados

Se detectaron varios problemas en varios puntos del sistema: en las gramáticas, en el uso de estructuras de control y funcionalidades de Perl no utilizadas, redundancias en el código, en la entrada y la salida de datos, etc.

En las diferentes gramáticas se detectaron los siguientes problemas:

1. Las diferentes gramáticas se utilizan en diversos módulos sin embargo el sistema carga del disco duro la gramática cuando la requiere lo cual es ineficiente y debería quedar cargada desde el inicio del programa para su disposición de los diferentes módulos.
2. La gramática de categorías gramaticales contiene, además de la definición de las construcciones con etiquetas POS, una variable que define el inicio de línea; si bien esta variable puede ser utilizada para pasar información desde la línea de entrada hasta el CD de salida, es restrictiva y es utilizada a lo largo del programa para detectar el inicio de una línea de trabajo, considerando que Perl ya tiene dentro de sus expresiones regulares un operador que tiene esta función.

(01_gramaticaPOS.pm 19)

```
our $inicioLinea = "<doc_codi [^ ]*?>:";
```

3. La gramática de categorías gramaticales, que es un módulo de Perl, ejecuta un código que marca los tiempos verbales en palabras etiquetadas como verbos. Dicha ejecución debe hacerse en un módulo del sistema y no en la gramática.
4. La gramática de patrones verbales contiene una variable que controla si las raíces (expresiones regulares que permiten la definición de formas verbales) son consideradas únicamente como texto o como expresión regular; esta variable fue agregada al desarrollo por Alarcón (2009) para la corrección de posibles errores. La variable siempre debe estar activada para que busque los verbos con una expresión regular.

El siguiente es un fragmento de la gramática donde se define si la raíz es o no una expresión regular:

(Archivo: 02_gramPVDs.txt 22-24)

 # Aquí se definen si la raíz va o no seguida de cualquier carácter, excepto espacio en blanco ([^]*?) hasta el siguiente "/"

 # Se expresa en valores SI - NO

<raizRegex>si</raizRegex>

5. Las expresiones regulares que sustituyen los elementos de la gramática de filtros, en el módulo (Archivo: 04_filtro.pm) se cargan en memoria cada vez que se lee una regla del filtro, y estas expresiones son iguales en todas las reglas del filtro, por lo que se tienen que inicializar sólo una vez en el proceso.

En el código se detectaron otros problemas:

1. En el módulo de evaluación si bien la evaluación es automática, las reglas que se utilizan no pueden ser modificadas ni existe una gramática que las contenga, por lo que no se pueden agregar nuevas reglas, además existen varios tipos de reglas en los módulos cuya modificación podría ser de interés.
2. Se graba y lee un archivo para la comunicación interna entre cada uno de los archivos, lo que propicia que el sistema sea más lento y agregue tiempo de procesamiento en la entrada y salida a disco duro.
3. El código está escrito utilizando las estructuras de control ineficientemente, haciendo difícil la lectura del código y agregando líneas innecesarias o redundantes.

Por ejemplo: (Archivo: 03_td.pm 106-115).

```
1 #Si el candidato contiene NEXO
2 if (/<nx_[^ ]*?>/i)
3 {
4 }
5 # Si el candidato NO contiene NEXO
6 # ANOTA
7 else
8 {
9       if (/lema=\"$lemaDef\"/i)
10      {
11              s/($inicioLinea)/<tipoD=\"$tipoDef\"\/>$1/gi;
12      }
}
```

Aquí se ve como las líneas 2, 3 y 4 plantean una estructura condicional; sin embargo, es estéril su producción; su caso *default* (*else*) línea 7 a 13, es el que genera la sustitución. Adicionalmente el condicional de 9-12 sólo ejecuta una sentencia por lo que las llaves sobran. Además cabe mencionar que los comentarios en 5 y 6 interrumpen la lectura de la estructura del condicional. Esto debe ser arreglado para verse así:

```
 #Si el candidato contiene NEXO
2 if (! /<nx_[^ ]*?>/i) # Si el candidato NO contiene NEXO, Anota
3 {
4       s/($inicioLinea)/<tipoD=\"$tipoDef\"\/>$1/gi if (/lema=\"$lemaDef\"/i);
5 }
```

Se comprime la misma sustitución en 5 líneas en lugar de las 13 anteriores. Los comentarios no rompen la lectura de la sentencia y se comprime el segundo condicional, ya que sólo ejecuta una sola sentencia. Esto no ahorra ciclos de procesamiento, pero sí líneas de código difícil de leer.

4. Debido a que el árbol de decisión en el "Archivo: 05_arbol.pm" utiliza extensivamente estructuras condicionales, se genera un código confuso debido al mal aprovechamiento de las propias estructuras y de las expresiones regulares usadas en los condicionales. Un ejemplo de esto es (Archivo: 05_arbol.pm:725-739):

```
1 if (/<izq>NULL<\/izq>/i)
2 {
3 }
4 elsif (/<izq>$demos.*?<\/izq>/i)
5 {
6 }
7 elsif (/<izq><pvd/i)
8 {
9 }
10 else
11 {
12   s/(<izq>|<.izq>)//gi;
13   s/<der>/<izq>/gi;
14   s/(<izq>.*?) (<pvd.*?>.*?<.pvd.*?>) /$1<\/izq> $2 <der>/gi;
15 }
```

Debiendo quedar así:

```
1       if (! /<izq>((NULL|$demos.*)<\/izq>|<pvd)/i)
2       {
3                       s/(<izq>|<.izq>)//gi;
4                       s/<der>/<izq>/gi;
5                       s/(<izq>.*?) (<pvd.*?>.*?<.pvd.*?>) /$1<\/izq> $2 <der>/gi;
6       }
```

Lo que ahorra más de la mitad de las líneas de código haciendo el código más legible y de fácil lectura.

5 El módulo de reetiquetado (Archivo: 06_retagging.pm) idealmente no debería existir ya que el etiquetado pertinente debería hacerse en cada uno de los módulos y no necesitaría arreglarse el etiquetado.

6 Existen ciclos de procesamiento continuos uno después de otro cuya condición de iteración es idéntica y es posible combinarlos en un solo ciclo.

3.1.3.3. Soluciones propuestas

Las líneas de trabajo que se plantean para presentar un módulo más flexible y siguiendo los problemas detectados se enumeran a continuación.

Para las gramáticas se propone:

1. Crear un módulo nuevo que inicialice todas las variables globales y cargue en la memoria, de una sola vez, todas las gramáticas a utilizar para evitar repetir dicho proceso; además de permitir que el "Archivo: 01_gramaticaPOS.pm" sea exclusivamente un módulo de definición de variables derivadas de las etiquetas POS.
2. Eliminar la etiqueta de inicio de línea tanto en la gramática (Archivo: 01_gramaticaPOS.pm) como en todos los módulos subsecuentes en donde se emplea y, después, reemplazarla por el operador de inicio de línea de Perl en expresiones regulares "^".
3. Eliminar de la gramática de patrones verbales (Archivo: 02_gramPVDs.txt) la variable de prueba de expresiones regulares en las raíces verbales, fijando su valor en sí en los módulos correspondientes donde es usada.
4. Crear una gramática de reglas de evaluación.

Para el código en sí en los diferentes archivos se propone:

1. Analizar el código de cada uno de los módulos para quitar el uso de producciones inútiles en las estructuras de control, además de buscar expresiones regulares redundantes o que puedan ser simplificadas.

2. Eliminar toda lectura y escritura hacia el disco duro, en los módulos principales del sistema, es decir, pasar la información por memoria y no por archivo.
3. Analizar el funcionamiento de reetiquetado y tratar de incluirlo en las expresiones regulares de los módulos previos y posteriores.

3.1.4. Salida del Sistema

El sistema no posee una salida estructurada, originalmente entrega los CDs ordenados por su evaluación y en un archivo de texto uno por línea y con las partes que los conforman etiquetados e informando al usuario cuántos se obtuvieron.

3.1.4.1. Problemas detectados

La salida no es flexible, el usuario no puede recuperar la información que requiere ni permite hacer operaciones posteriores con los CDs. Además de que no permite la limpieza y filtrado de los CDs y sus etiquetas. Adicionalmente los usuarios pueden requerir como salida únicamente el texto segmentado de la entrada o solamente etiquetado con POS.

3.1.4.2. Soluciones propuestas

Implementar un módulo de salida que permita comunicar al usuario con la información recuperada por Ecode, que permita presentar los CDs al gusto del usuario y que sea capaz de comunicarse con los demás módulos así como con las interfaces de usuario.

El módulo tendrá la capacidad de procesar los CDs, limpiarlos y agruparlos por término semejante además de permitir conservar o eliminar etiquetas y presentar los CDs con un procesamiento posterior, todo en función de la facilidad de uso y de las salidas requeridas.

3.2. Empleo del Ecode.

Desde su desarrollo se ha propuesto al sistema Ecode como motor de más aplicaciones que pueden aprovechar a los contextos definitorios, entre las que destaca el Describe®. Por su parte, el Describe® es un sistema que permite buscar en internet documentos que contengan un término dado y PVDs; obtiene, por medio del Ecode, las definiciones de dicho término, además de presentar las definiciones agrupadas por posible homonimia. Este desarrollo se utiliza también como herramienta académica y de apoyo dentro del Grupo de Ingeniería Lingüística (GIL).

3.2.1. Página del Ecode.

Cuando Alarcón (2009) desarrolló el Ecode, a la par se creó una página descriptiva de lo que es Ecode en donde se tenía pensado crear una aplicación web del mismo; dicha posibilidad no llegó a materializarse. Se subieron análisis de un corpus en donde se tomaron como punto de partida los términos simples del Vocabulario Básico del Genoma Humano del Corpus Técnico del IULA (http://www.iula.upf.edu/rec/vbgenoma/esp/frames.html). Se extrajeron los CDs de textos de términos del genoma humano y se agruparon por tipo de definición y además por PVD, indicando cuántos CDs se extrajeron y cuántos candidatos fueron filtrados.

Tipo de Definición	Patrón Verbal	CDs	No Relevantes
Genus y Diferencia	ser + det + N	7	52
Genus y Diferencia	definir como	5	5
Genus y Diferencia	entender como	1	4
Genus y Diferencia	concebir como	0	0
Genus y Diferencia	identificar como	0	32
Extensional	constar de	3	11
Extensional	formar por	7	27
Extensional	contener	1	4
Extensional	Tener	2	12
Funcional	Usar en	0	17
Funcional	Usar como	0	8
Funcional	Usar para	0	18
Funcional	Utilizar en	1	41

Tabla 3.10 Resultados de GEN por Tipo de definición y PVD en la página de Ecode (http://www.iula.upf.edu/rec/vbgenoma/esp/frames.html)

Además permite consultar los contextos extraídos de cada término por PVD. Cabe mencionar que dichos resultados fueron procesados y sólo se subió una interfaz para consultarlos, de ninguna manera está ligado el sistema Ecode con dicha página y fue pensada únicamente para informar y difundirlo[11].

3.2.2. Describe

Describe es un proyecto desarrollado en el Grupo de Ingeniería Lingüística, patrocinado a través de un proyecto CONACyT, que lleva el nombre de "Extracción de relaciones léxicas para dominios restringidos a partir de contextos definitorios en Español".

El sistema Describe consiste en un conjunto de módulos que, a través de una interfaz web, reciben un término y lo buscan por medio de un motor de búsqueda incluyendo el término y los patrones verbales definitorios. De los URLs resultantes de la búsqueda, se descarga el texto de los documentos.

[11] Su URL es: http://brangaene.upf.es/ecode

Ecode es el motor principal de Describe, utiliza los documentos extrayéndoles el texto, se obtienen los CDs y se filtran por término para poder entregar un conjunto de CDs agrupados por definición.

Después, los contextos extraídos se agrupan, en otro módulo, por similitud de las definiciones y se presentan al usuario en una interfaz web.

El Ecode cumple la función de núcleo del proyecto Describe; si bien en este momento los módulos que constituyen a Describe no se encuentran plenamente integrados, el Ecode es parte integral de éste.

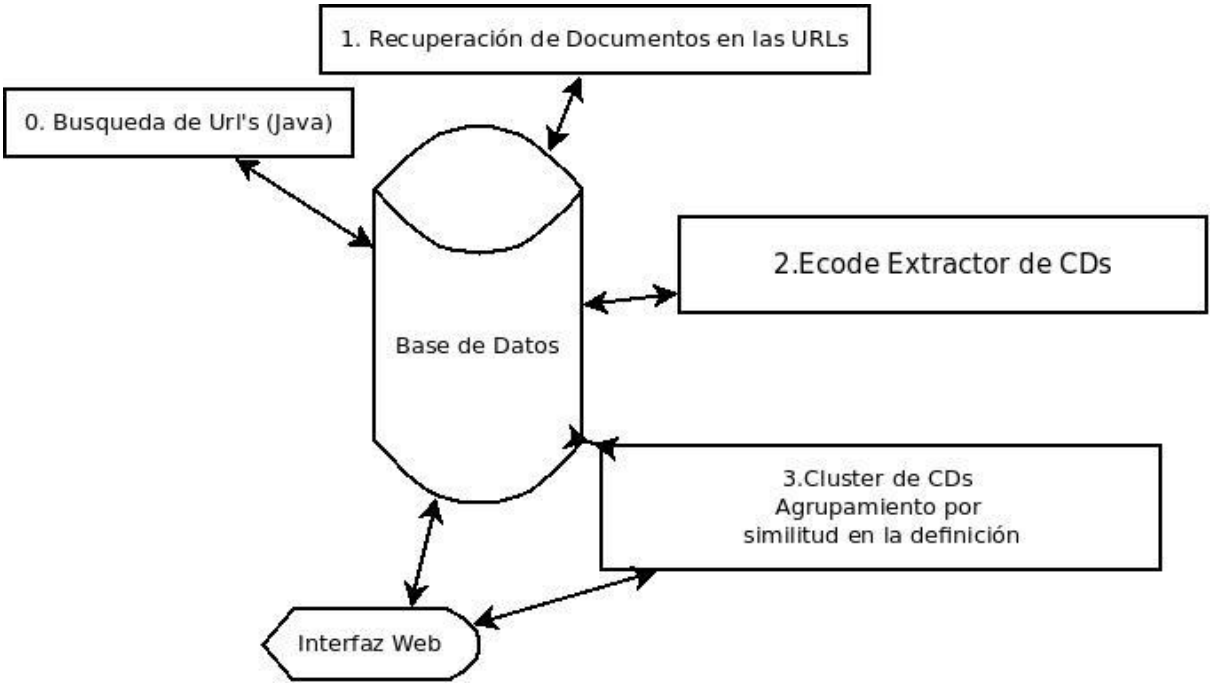

Figura 3.7 Esquema del sistema Describe

El flujo de un término hasta encontrar sus CDs y presentarlos al usuario final se representa en la siguiente figura.

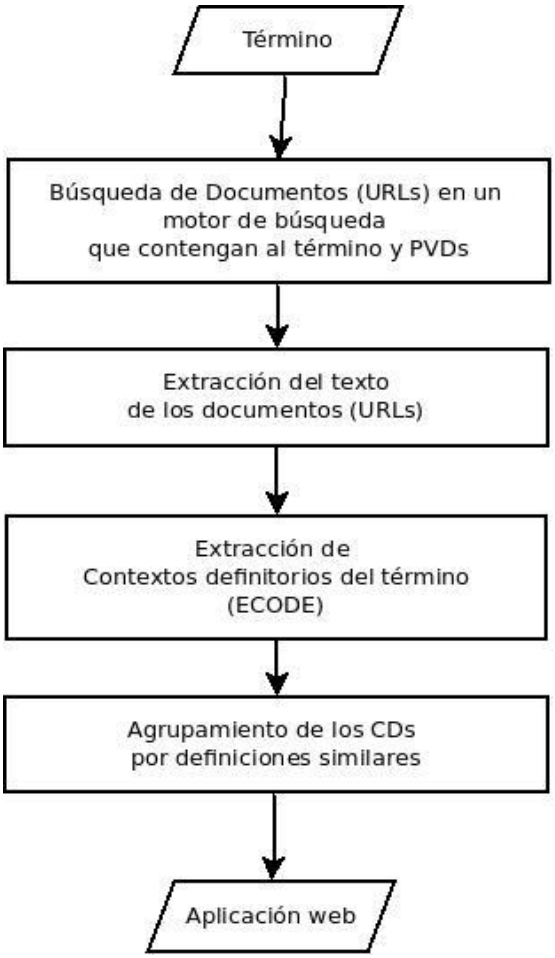

Figura 3.8 Flujo del Sistema Describe

Describe tiene su propio sitio web[12] que cuenta con alrededor de 500 términos agrupados y listos para ser presentados.

3.3. Recapitulación de problemas y líneas de trabajo

En general se puede decir que el sistema Ecode presenta 3 problemas principales:

1. No es flexible en la entrada y requiere de un preprocesamiento que no es sencillo y no tiene las herramientas adecuadas.

[12] http://www.describe.com.mx:8080/describeRe/15/

Se propone el desarrollo de un módulo de entrada que preprocese el texto, lo separe por oraciones, lo etiquete con sus categorías gramaticales, haga un filtrado previo y presente una entrada lista para Ecode.

2. Los módulos originales presentan varios problemas: primero, las estructuras de control no fueron utilizadas adecuadamente y existen archivos intermedios entre los módulos lo cual lo hace más lento. Además tiene una gran cantidad de variables globales que, o se usan una sola vez, o ya fueron declaradas con otro nombre en otro módulo. Por otra parte, las gramáticas presentan variables innecesarias o que no requieren formar parte de ellas, además de que se necesita de la posibilidad de poder agregar más reglas a ellas.

La solución que se propone es reescribir el código en el que las estructuras de control no se utilizaron correctamente, eliminar los accesos a disco duro intermedios entre los módulos, y eliminar variables excesivas o declaradas como globales.

Modificar las gramáticas, limpiándolas y permitiendo únicamente la existencia de variables estrictamente requeridas, además de admitir que el usuario agregue reglas en todas y cada una de ellas.

3. La salida del sistema únicamente es una lista de contextos y no contiene ningún procesamiento posterior (agrupamiento, lematización, conexión a bases de datos, limpieza de etiquetas, salida en XML o una aplicación web).

Para proveer funcionalidades que permitan dar formato a los CDs de salida es necesario crear un módulo de salida que permita hacer una serie de operaciones para la interconexión con otros sistemas y la presentación clara al usuario, así como una aplicación web.

Estos son los principales problemas y las propuestas básicas para optimizar y desarrollar el sistema Ecode. En el siguiente capítulo se tratarán las líneas de trabajo planteadas y sus soluciones implementadas así como su posible desarrollo futuro.