

**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUDIRECCIÓN ACADÉMICA**

No de Serie: TT0311

Serie: Amarilla
Trabajo Terminal

Mayo 2002

**SISTEMA DE MIMETISMO BASADO EN
GRAMÁTICA PARA OCULTAMIENTO DE
INFORMACIÓN**

Fátima Margarita Lechuga Blanco

e-mail: lechugafresca@hotmail.com
cisne_x@latinmail.com

Dirección: Calle Guillermo Prieto No.15 Col. La Mora
Ecatepec Edo. México CP.55030
Teléfono: 57-87-30-88
57-87-33-76

Mario César Lima Rodríguez

e-mail: mario_lima_rgz@hotmail.com
loco_maniatico@hotmail.com

Dirección: Oriente 166 No. 338 Col. Moctezuma
2da Sección. México, D.F. CP 15530
Teléfono: 55-71-93-64
55-71-27-49

Resumen

En este trabajo se presenta una forma de ocultar y recuperar mensajes dentro de texto apoyándose en gramáticas libres de contexto, para lo cual se desarrollo un sistema ocultador que genera un texto en el cual se encuentra el mensaje oculto y un sistema extractor que recupera el mensaje oculto. La solución planteada se basa en la aplicación de la lingüística matemática, compiladores, seguridad informática.

Palabras clave: Gramática Libre de Contexto, Esteganografía, Lingüística Matemática, Seguridad informática, Caminos de Markov, Generadores de textos, Manejadores de gramáticas, Criptografía, Ocultamiento de texto en texto, Matrices de frecuencias acumuladas.

INDICE

CONTENIDO	PÁGINA
INTRODUCCIÓN	3
CAPÍTULO 1 MANEJADOR DE GRAMÁTICAS	6
1.1 Manejo Léxico	6
1.2 Manejo Sintáctico	9
1.2.1 Manejador de gramáticas	9
CAPÍTULO 2 GENERADOR DE TEXTO	17
2.1 Generador de matriz de frecuencias	20
2.2 Generador de Matrices de frecuencias acumuladas horizontal	21
2.3 Generador de texto	22
2.4 Generador de Matrices de frecuencias acumuladas vertical	28
2.5 Generador de textos para ocultar mensajes	29
CAPÍTULO 3 MÓDULOS DEL SISTEMA	36
3.1 Módulo Ocultador	36
3.1.1 Características	36
3.1.2 Algoritmo	36
3.1.3 Manejador de gramáticas para el ocultador de texto	38
3.2 Módulo Extractor	39
3.2.1 Algoritmo	39
3.2.2 Manejador de gramáticas para el extractor de texto	40
CAPITULO 4 RESULTADOS.	42
4.1 Ejemplo 1.	42
4.2 Ejemplo 2.	49
4.3 Ejemplo 3.	53
4.4 Ejemplo 4.	56
CONCLUSIONES	63
ANEXOS	64
ANEXO A CRIPTOGRAFÍA	65
ANEXO B GRAMÁTICAS GENERATIVAS	66
FUENTES DE INFORMACIÓN	69

INTRODUCCIÓN

Muchos aspectos de la seguridad informática están cubiertos por la criptografía, por los sistemas de respaldo y por los lineamientos y precauciones que se toman en las instituciones, como por ejemplo guardias de seguridad en donde se encuentra información importante, higiene en el trabajo cerca de las computadoras (bebidas, alimentos), manejo de fuentes alternas de respaldo y energía.

Al igual que las anteriores técnicas y métodos, nuestro trabajo terminal cubre aspectos de seguridad, pero mediante aplicación de la *esteganografía, la cual es la ciencia ó arte de ocultar un mensaje dentro de otro de tal forma que se oculte la existencia del mensaje original*.

El término "esteganografía" viene del griego stegos, que significa "cubierta", por lo que esteganografía significaría "escritura oculta" o "escritura encubierta". Así pues, la esteganografía es el conjunto de técnicas que nos permiten ocultar o camuflar datos. [pág. web3]

La esteganografía es muy antigua, por lo que se tienen muchas técnicas esteganográficas para ocultar información, como por ejemplo, escribir en una hoja un mensaje con jugo de limón y entregarlo a la persona a quien va dirigido, la cual pasa muy ligeramente una flama por debajo de esta hoja, con lo que, puede enterarse del mensaje oculto.

Otro ejemplo muy usado de aplicación esteganográfica es por ejemplo la forma en que hablamos muchas personas (en especial los mexicanos) en doble sentido, mientras estamos platicando con un grupo de personas sobre un tema, a la vez estamos platicando con nuestro amigo acerca de un tema al cual las demás personas no tienen acceso, solo nuestro amigo y nosotros.

Una aplicación de la esteganografía a la computación, es por ejemplo cuando una imagen, se reduce hasta que esta del tamaño de un punto, el cual pegamos dentro de un texto, indicando en que punto se encuentra la imagen a la persona a quien va dirigida dicha información.

Una de las características principales de la esteganografía es que para conservar intacta y oculta nuestra información trata de engañar o aprovechar nuestros sentidos, por ejemplo, cuando ocultamos un mensaje en una imagen, engaña a nuestros ojos, ya que, nuestros ojos no pueden diferencia entre una imagen sin mensaje y una con mensaje, dado que, no notan pequeños píxeles diferentes o la diferencia de tono de los colores de una imagen con otra. Lo mismo sucede al ocultar un mensaje en un archivo de sonido, nuestro oído no percibe ese pequeño "ruido" entre un archivo de sonido y otro.

La esteganografía sin duda es empleada desde mucho tiempo atrás, pero ahora con la ayuda de la Informática y la Computación se ha vuelto, al igual que la criptografía, una herramienta muy poderosa para mantener la confidencialidad e integridad de nuestra información.

Dentro de las aplicaciones de la esteganografía más comúnmente empleadas están las que ocultan texto en imágenes, imágenes en imágenes o sonido en sonido, como por ejemplo:

Stego. El cual es un paquete que permite ocultar pequeños archivos en imágenes de tipo gif., se encuentra en la página www.alakarga.es.org/estega.html

S-Tools4. Esta es una herramienta que nos permite ocultar datos en archivos tipo wav, bmp y gif., se encuentra en linkbeat.com/lb/files/s-tools4.zip

MP3Stego Este es un paquete que oculta archivos comprimidos y encriptados en archivos de sonido MP3., se encuentra en www.cl.cam.ac.uk/~fapp/steganography/mp3stego

Texto, aparece en www.arnal.es/free/cripto/estega/estegind.htm y transforma archivos codificados con unencode a archivos de texto que contienen frases en inglés.

Dentro de este contexto, *específicamente en nuestro trabajo buscamos ocultar y recuperar textos en textos apoyándonos en gramáticas.*

Nuestro sistema tiene como entradas textos de ejemplos, una gramática y un mensaje. De estas entradas obtenemos como salida un texto con un mensaje oculto y por otro lado a partir de la misma gramática y el texto con el mensaje oculto, obtenemos como salida el mensaje original. (Fig. 1).

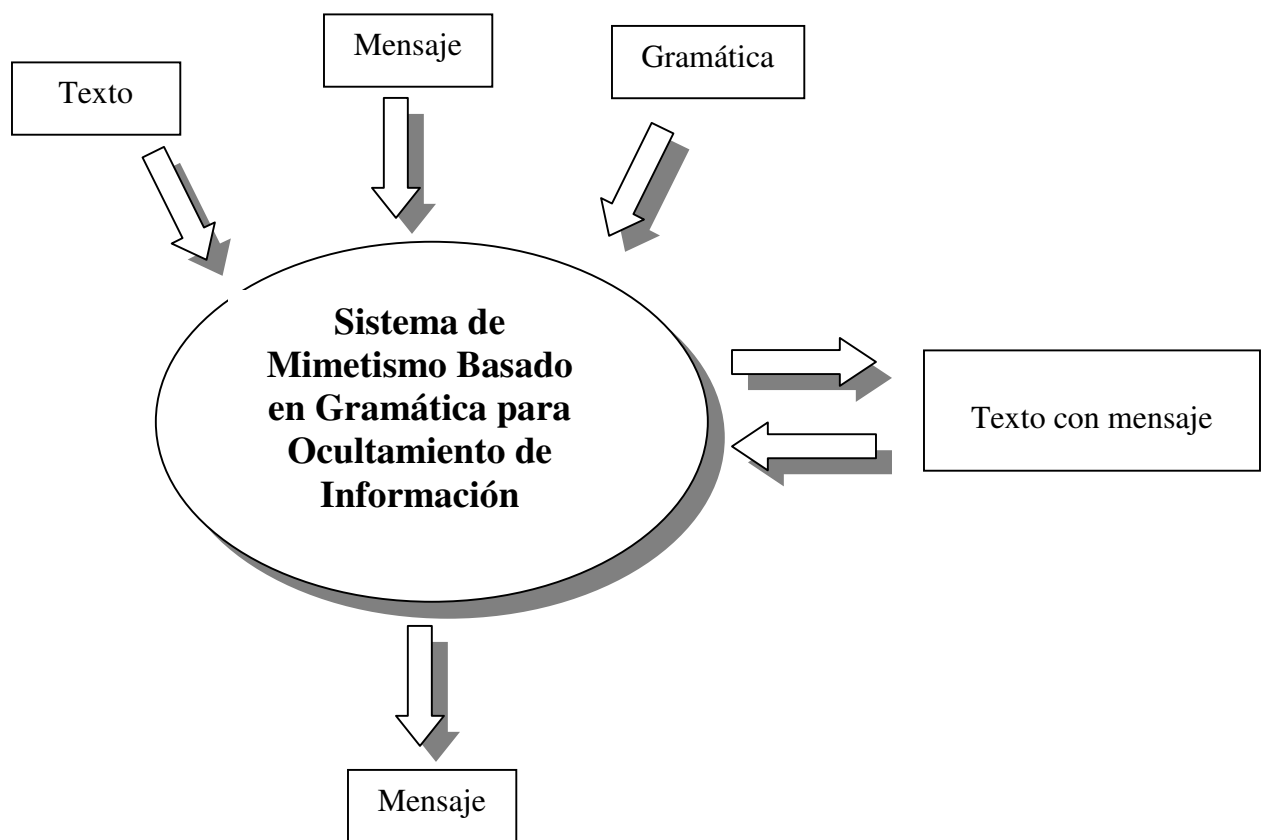


Fig.1.Diagrama de Contexto

Las aplicaciones esteganográficas se caracterizan entre otras cosas por estar formadas de dos grandes subsistemas o bloques: Un ocultador, que se encarga de esconder la información y un extractor, que es el que se encarga de realizar el proceso reversible. Nuestro sistema no es la excepción, como se muestra en la Fig. 2 nuestro sistema tiene 2 módulos, el Ocultador y el Extractor.

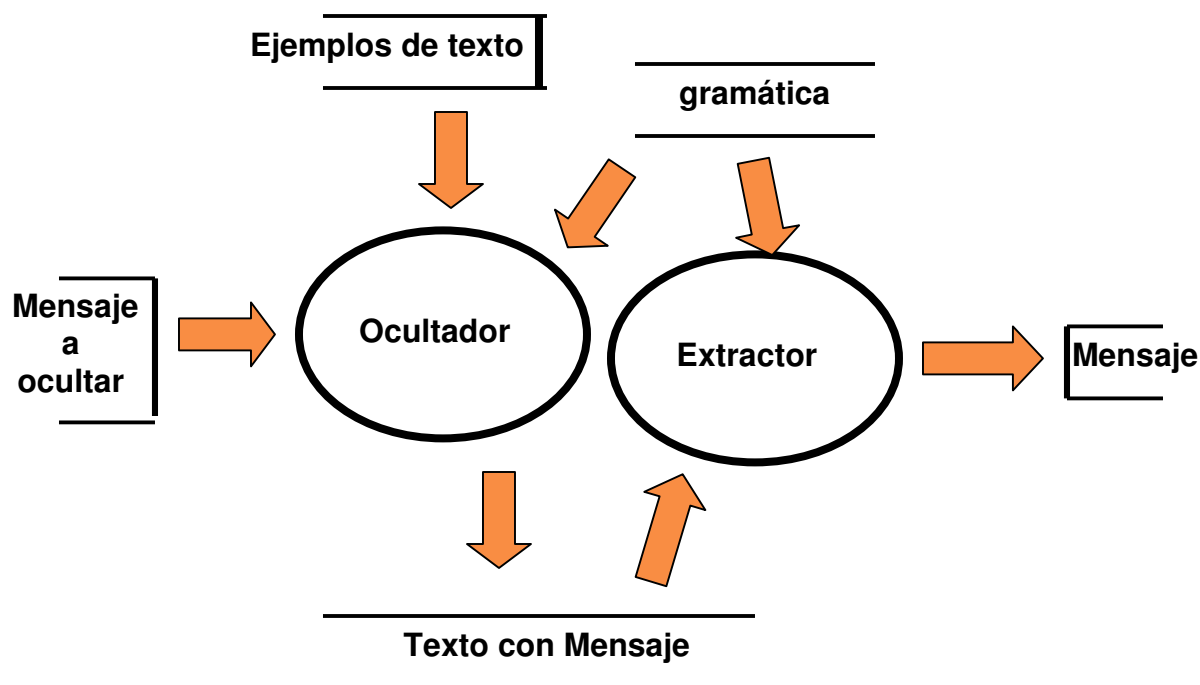


Fig.2 Diagrama General del Sistema

El módulo ocultador tiene como entradas la gramática, textos de ejemplo y el mensaje a ocultar, obteniendo como salida un texto con un mensaje oculto. El propósito de este proceso es ocultar los elementos del mensaje “secreto” dentro de un texto, siguiendo la estructura de la gramática, de tal manera que nos de como resultado un nuevo texto con el mensaje ya incrustado.

Por su parte el módulo extractor tiene como entrada el texto con el mensaje oculto, la gramática (que es la misma que la del ocultador) de la cual se ayuda para extraer el mensaje oculto que obtenemos como salida. Este proceso tiene como propósito el detectar los elementos en el texto, siguiendo la estructura de la gramática, que fueron incrustados en el proceso de ocultamiento, para poder así recuperar el mensaje secreto.

Además para poder desarrollar el sistema nos apoyamos en un Manejador de Gramáticas y un Generador de Textos, los cuales describimos a continuación.

CAPÍTULO 1 MANEJADOR DE GRAMÁTICAS

Como se ha comentado el objetivo principal es desarrollar un sistema que nos permita ocultar un mensaje en un texto, siguiendo las estructuras de una gramática. Para atacar este problema primero desarrollamos una serie de herramientas generales para el manejo léxico y sintáctico de nuestro sistema, las cuales mostramos a continuación.

1.1 Manejo Léxico

El análisis léxicográfico o reconocimiento de componentes léxicos, lee y agrupa cadenas de caracteres que se encuentran en el archivo fuente para formar componentes léxicos. [Lemone].

En particular en nuestro sistema, el manejo léxico se lleva a cabo mediante un manejador de unidades léxicas, que toma una lista de palabras con su tipo léxico almacenadas en un archivo y genera una tabla de unidades léxicas. En el archivo se encuentra un conjunto de palabras del idioma español seguida cada una de ellas de su clasificación o tipo, es decir si esa palabra es un sujeto, un pronombre, articulo, etc. Por ejemplo si se tiene el archivo de palabras tipos siguiente:

Archivo de palabras y tipos

La p niña s juega v en c el p patio l con c gato s y c un c perro s

Donde:

v verbo
s sujeto
p pronombre
c conjunción
etc.

La tabla de unidades léxicas quedaría:

Palabras	Tipos
La	p
niña	s
juega	v
en	c
el	p
...	...
...	...

Como se muestra en la Fig.1.1 la tabla de unidades léxicas entra, junto con un texto a un proceso, el cual toma las diferentes palabras del texto y las compara con las que se encuentran en la tabla de palabras, obteniendo a que tipo pertenecen las palabras de texto y entregando esto como resultado.

En el ejemplo de la Fig.1.1, si el texto de entrada es:

El gato come

El resultado es

p s v.

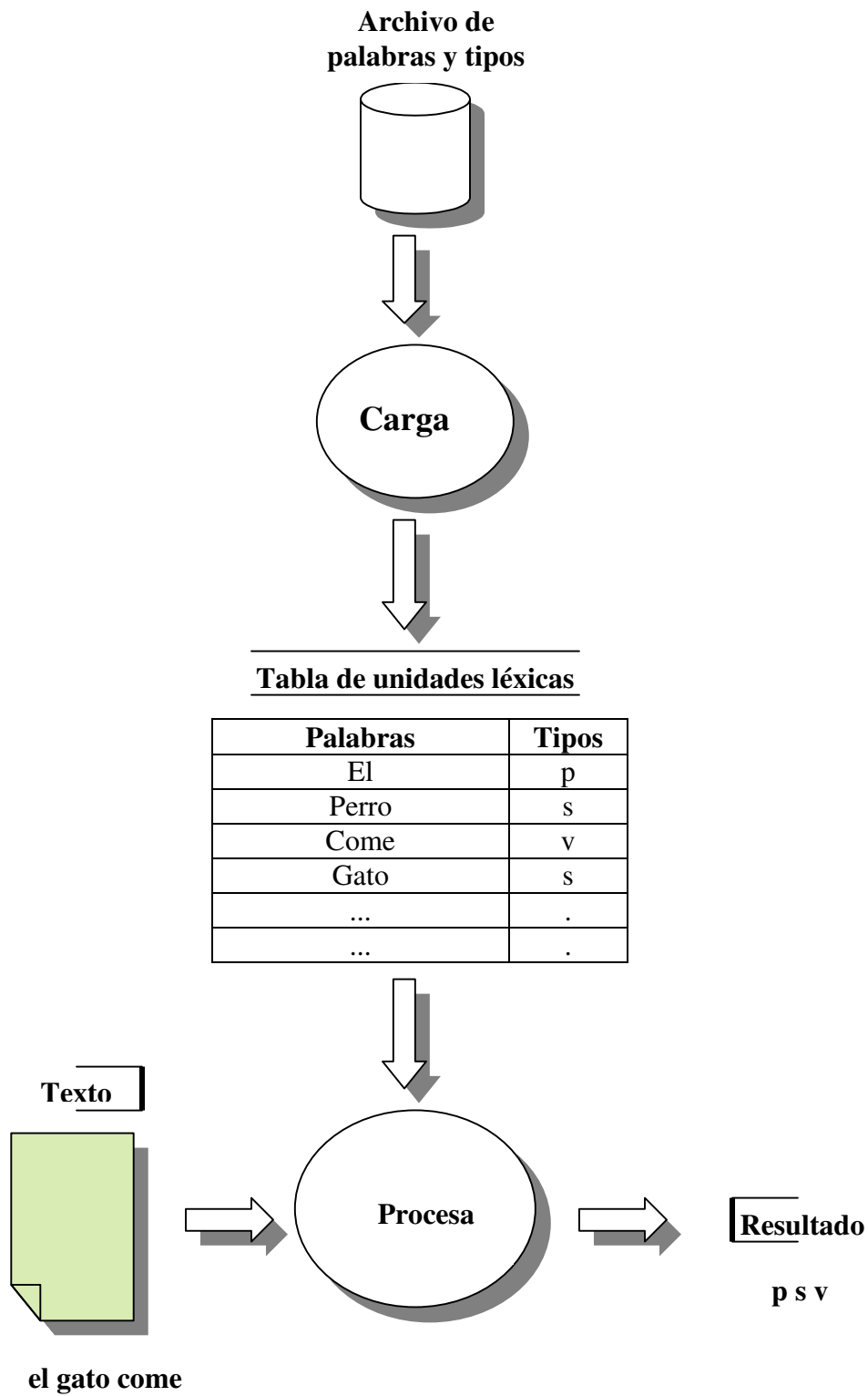


Fig.1.1 Manejador de unidades léxicas

1.2 Manejo Sintáctico

El análisis gramatical, también conocido como *análisis sintáctico*, es la fase que intenta agrupar secuencias de unidades léxicas o tokens en categorías sintácticas [Lemone].

Y se refiere a la forma como se relacionan las unidades léxicas o palabras dentro del texto, es decir la estructura o sintaxis que siguen y que esta determinado por la gramática, que es el conjunto de reglas que representan un lenguaje.

El análisis sintáctico puede ser de manera general de dos tipos:

Análisis sintáctico descendente

Análisis sintáctico ascendente

El análisis descendente se lleva a cabo para gramáticas LL, donde la primera L indica que se va analizando la cadena de entrada de izquierda a derecha y la segunda L indica que se va derivando el elemento más izquierdo de la producción gramatical.

Por su parte, el análisis sintáctico ascendente se tiene para las gramáticas LR, donde L indica que se analiza la cadena de entrada de izquierda a derecha y la R indica que se va derivando de derecha a izquierda.

En este trabajo se desarrollo un manejador de gramáticas, que se encargará del manejo sintáctico, de tipo descendente recursivo, ya que, conforme avanza en el análisis se llama recursivamente cada que encuentra un elemento no Terminal en la gramática.

Ahora bien la forma como se integra el manejo de la gramática en el Ocultador, es diferente a su implementación en el Extractor, por lo que, en este subcapítulo se mostrará como se construyó el manejador de gramáticas independiente de su aplicación y posteriormente en los capítulos correspondientes se mostrará su integración tanto en el Ocultador como en el Extractor.

1.2.1 Manejador de gramáticas

El manejador de gramáticas de manera general obtiene un texto de entrada y verifica si ese texto es parte del lenguaje generado por una gramática y dependiendo de si el texto pertenece o no a ese lenguaje, entrega un resultado verdadero o falso.

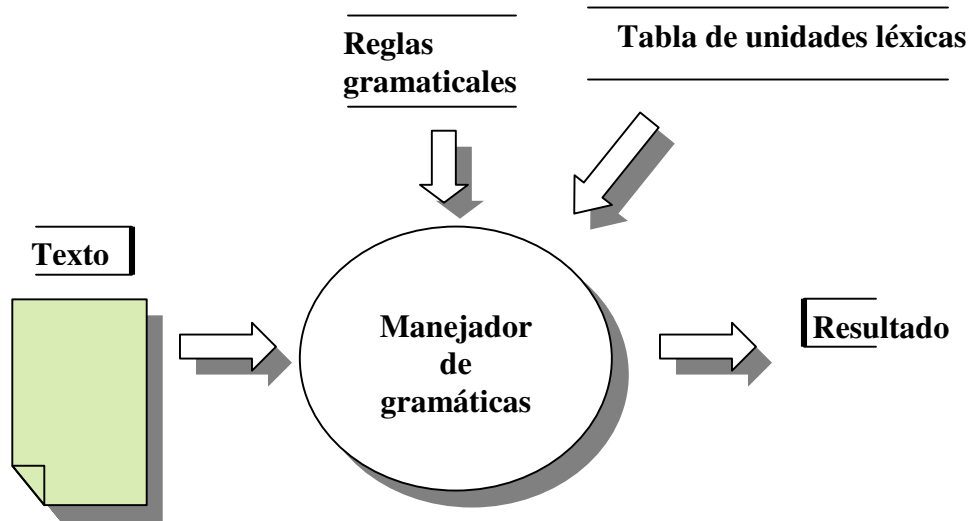


Fig. 1.2. Manejador de gramáticas

1.2.1.1 Versión 0 del Manejador de Gramáticas

Nuestra primera versión de manejador de gramáticas es muy sencilla, y analiza cadenas gramaticales que están formadas por puros elementos terminales y corresponden a la parte derecha de una regla gramatical, como por ejemplo la cadena p s v . Como se ve en la Fig. 1.3.

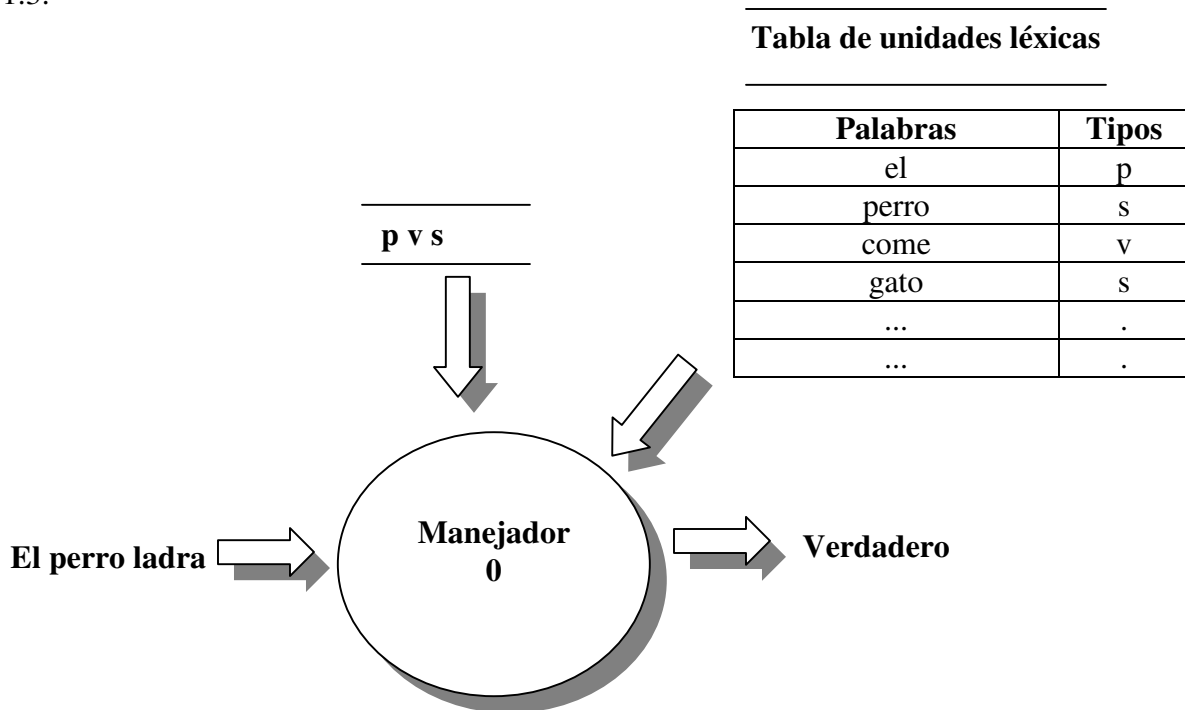


Fig. 1.3. Manejador de gramáticas versión 0

Donde:
 p representa pronombre
 s sujeto
 v verbo

El algoritmo se muestra a continuación:

Lee texto
 Mientras no haya un fin de cadena gramatical
 Toma elemento de la cadena gramatical
 Lee el elemento del texto
 Verifica que el tipo del elemento del texto sea el mismo que el de la cadena gramatical
 De no ser así, entrega falso
 Si ya termino el texto y no marco falso, entrega verdadero

Para explicar como funciona este manejador de gramáticas tomaremos el ejemplo *p s v*

Donde:
p representa pronombre
s sujeto
v verbo

y el texto de entrada
el perro ladra

En la tabla 1.1 se muestra el algoritmo propuesto anteriormente.

Elemento de la gramática	Palabra de texto	Tipo de la palabra de texto	resultado	Comentario
p	el	p	-	Tipo de la palabra y elemento de la gramática son iguales
-	Perro	-	-	-
s	perro	s	-	Tipo de la palabra y elemento de la gramática son iguales
-	ladra	-	-	-
v	ladra	v	-	Tipo de la palabra y elemento de la gramática son iguales
v	ladra	v	-	Gramática = fin y texto = fin
-	-	-	Verdadero	-

Tabla 1.1 Ejemplo de funcionamiento de la Versión 0 del Manejador de Gramáticas

1.2.1.2 Versión 1 del Manejador de Gramáticas

Ahora, ponemos la producción gramatical de manera más formal, aceptando reglas como $S \rightarrow psv$

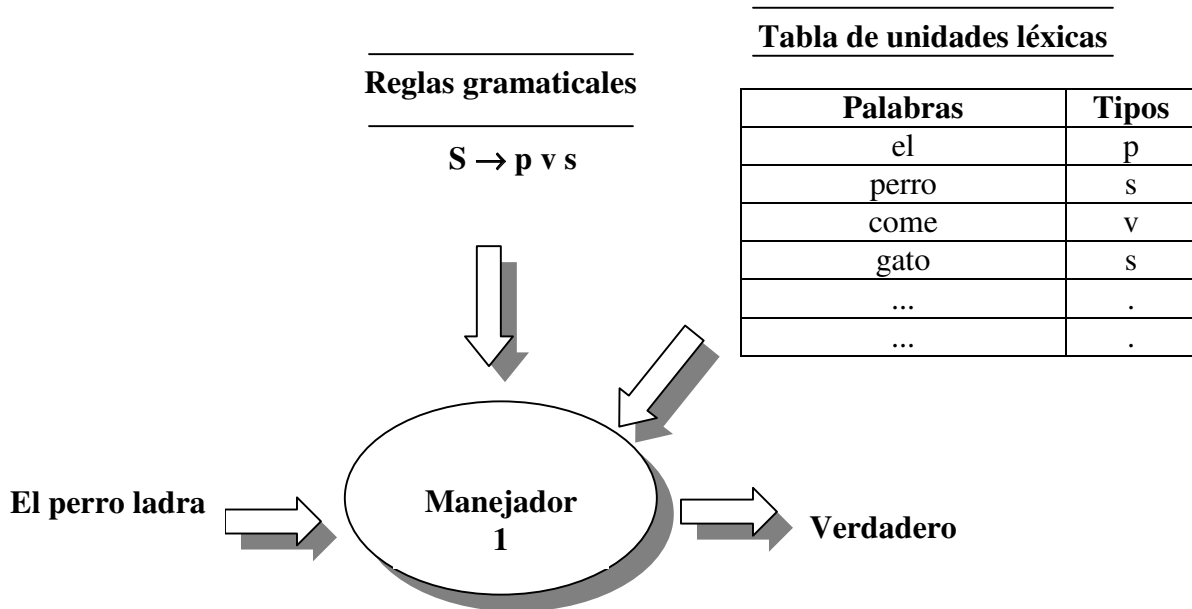


Fig. 1.4. Manejador de gramáticas versión 1

El algoritmo es casi el mismo que el de la versión 0, con excepción que primero verifica que la cadena empiece con el axioma de la gramática.

Lee la regla gramatical

Lee un texto

Verifica que el primer elemento de la regla sea el axioma

Lee el lado derecho de la producción

Mientras no haya un fin de producción

 Toma elemento de la producción

 Lee elemento del texto

 Verifica que el tipo del elemento del texto sea el mismo que el del elemento de la producción

 De no ser así, entrega falso

Si ya termino el texto y no marco falso, entrega verdadero

En la tabla 1.2 seguimos el algoritmo con la regla

$S \rightarrow p s v$

Y el texto a analizar

el perro ladra

Elemento de la gramática	Palabra de texto	Tipo de la palabra de texto	resultado	Comentario
S	-	-	-	Encontró S
p	el	p	-	Tipo de la palabra y elemento de la gramática son iguales
-	Perro	-	-	-
s	perro	s	-	Tipo de la palabra y elemento de la gramática son iguales
-	ladra	-	-	-
v	ladra	v	-	Tipo de la palabra y elemento de la gramática son iguales
v	ladra	v	-	Gramatica=fin y texto = fin
-	-	-	Verdadero	-

Tabla 1.2 Ejemplo de funcionamiento de la Versión 1 del Manejador de Gramáticas

1.2.1.3 Versión 2 del Manejador de Gramáticas

Ahora complicamos un poco la gramática para admitir varias reglas gramaticales, como por ejemplo:

$S \rightarrow ABC$

$A \rightarrow p$

$B \rightarrow s$

$C \rightarrow v$

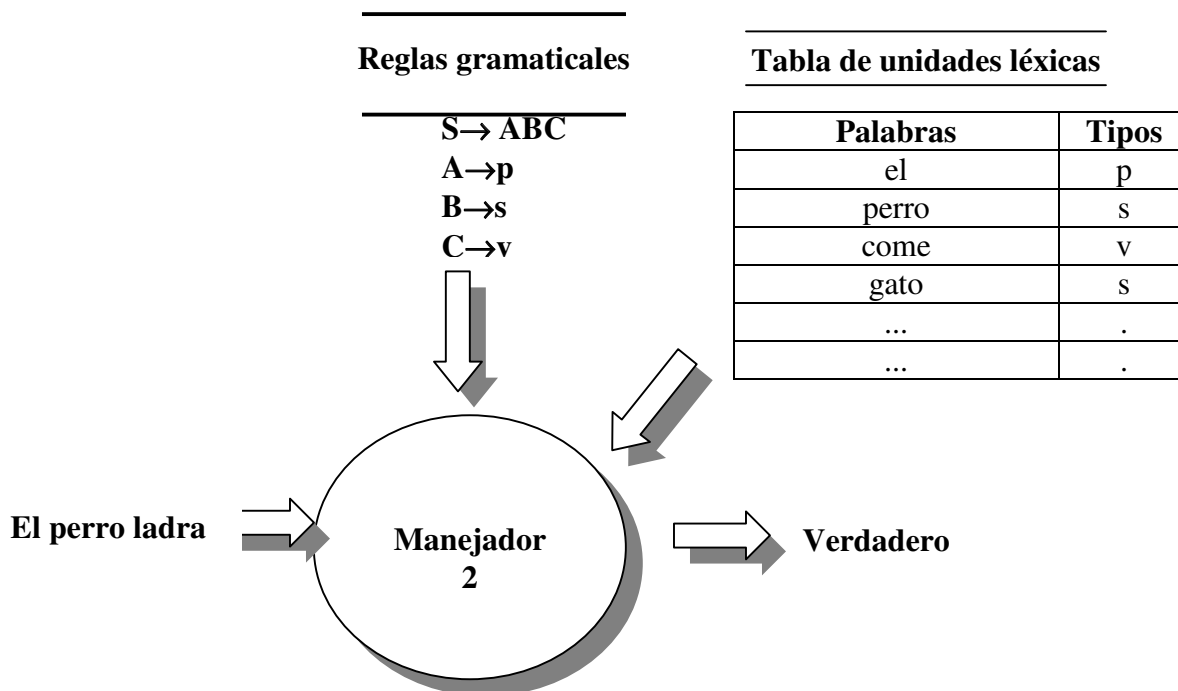


Fig. 1.5 .Manejador de gramáticas versión 2

En este caso se tiene que ver si el elemento dentro de la producción gramatical es o no un terminal. Por facilidad colocamos el código como una función llamada Manejador() y las reglas gramaticales se colocan en una tabla donde, cada renglón tiene una regla gramatical y el primer elemento del renglón es la parte izquierda de la regla y a partir del segundo elemento se coloca la parte derecha.

Por ejemplo si se quieren manejar las reglas gramaticales

$S \rightarrow ABC$

$A \rightarrow p$

$B \rightarrow s$

$C \rightarrow v$

En la tabla quedan como

S	A	B	C
A	p		
B	s		
C	v		

El algoritmo modificado se muestra a continuación:

```

Manejador (NoTerminal)
{
  Busca una regla gramatical que empiece con el NoTerminal
  Lee la regla gramatical
  Mientras no haya un fin de producción
  {
    Lee elemento de la producción
    Si el elemento es un terminal
      {
        Lee elemento del texto
        Verifica que el tipo de la palabra del texto sea el mismo que el elemento de la
        producción
        De no ser así, entrega falso
      }
    Si es un elemento no terminal
      Llama a Manejador(elemento no terminal)
  }
  Si ya termino el texto y no marco falso, entrega verdadero
}

```

En este caso el texto a analizar se pasa en forma global a la función.

1.2.1.4 Versión 3 del Manejador de Gramáticas

La siguiente versión del manejador de gramáticas ya permite manejar varias reglas para cada elemento no terminal, como por ejemplo:

$S \rightarrow abA$
 $S \rightarrow mX$
 $A \rightarrow c$
 $X \rightarrow xy$
 $X \rightarrow am$

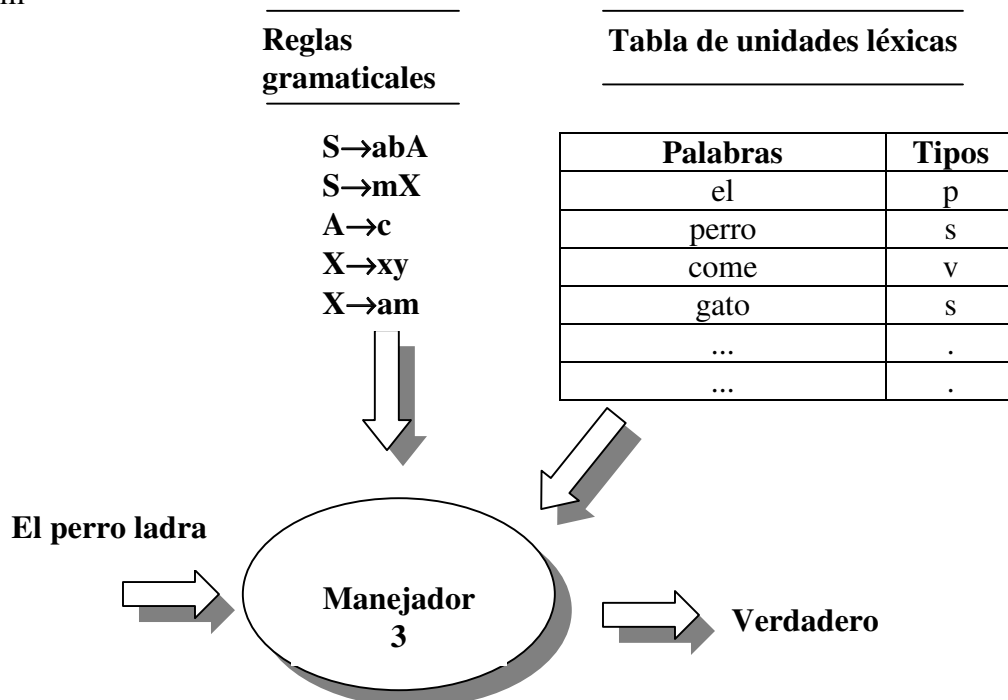


Fig. 1.6. Manejador de gramáticas versión 3

Para la cual el algoritmo queda modificado de la siguiente forma:

```
Manejador (NoTerminal)
{
  contador 0
  Encuentra la primera regla gramatical que empiece con NoTerminal
  bandera = 'F'
  Mientras existan reglas que empiezan con el mismo NoTerminal , la bandera = 'F' y contador =0
  {
    Lee la producción gramatical que empieza con el NoTerminal
    bandera = 'V'
    Mientras no haya un fin de producción y la bandera sea 'V'
    {
      Lee elemento de la producción
      Si el elemento es un terminal
      {
        Si el tipo del terminal es igual al tipo del elemento del texto
          Avanza elemento producción,
          Avanza elemento de texto
          contador++
        Si no
          bandera = 'F'
      }
      Si es un elemento no terminal
        Llama a Manejador(elemento no terminal)
        Si regreso falso del manejador
          bandera = 'F'
    }
    Lee siguiente producción
  }
  Regresa bandera
}
```

Esta versión del manejador de gramáticas permite manejar gramáticas libres de contexto, que tienen reglas con recursividad empotrada como:

```
S→(S)
S→x
```

CAPÍTULO 2 GENERADOR DE TEXTO

Uno de los problemas que se tienen en la esteganografía de textos es que las palabras a ocultar deben quedar inmersas en textos que no permitan detectarlas, por lo que, debemos buscar que las palabras que rodean a las palabras ocultas, sean palabras que normalmente aparecen junto a ellas.

Para lograrlo desarrollamos un generador de textos basado en caminos de Markov representados mediante matrices de frecuencias acumuladas, donde, para cada palabra indicamos cuales son las palabras que tienen mayor probabilidad de aparecer a su alrededor.

En la Fig.2.1 se muestra la estructura de un generador de texto, que a partir de textos en español genera una matriz de frecuencias acumuladas, con la que a su vez se genera texto.

Los Caminos de Markov representan procesos en los que el estado presente solo depende del estado anterior.

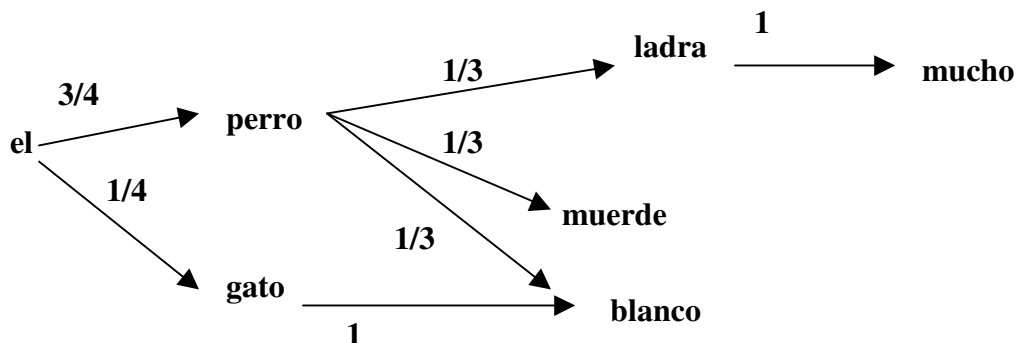
Es decir, en este caso cada palabra representa un estado diferente.

Por ejemplo, si tenemos el siguiente texto:

El perro ladra mucho
El perro muerde
El gato blanco
El perro blanco

La palabra 'El' es un estado, la palabra 'perro' es un estado diferente y así sucesivamente para cada una de las diferentes palabras del texto.

Y a partir del mismo ejemplo se genera el siguiente Camino de Markov:



Ya que si se ven las oraciones de ejemplo, la palabra 'el' aparece 4 veces y de estas 3 veces aparece la palabra 'perro', una vez aparece la palabra 'gato' después de 'el' y así para las demás palabras.

Estos valores de frecuencias son representados con la matriz de frecuencias, por ejemplo: Si tenemos el siguiente texto de ejemplo

el perro ladra en su casa.
 el perro blanco está en la casa.
 un perro está en el jardín.
 un gato blanco.

Este texto generaría la siguiente matriz de frecuencias.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	1	0
perro	0	0	1	0	0	0	1	1	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	1	0	0	0	1	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	1	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	1
.	2	0	0	0	0	0	0	0	0	1	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

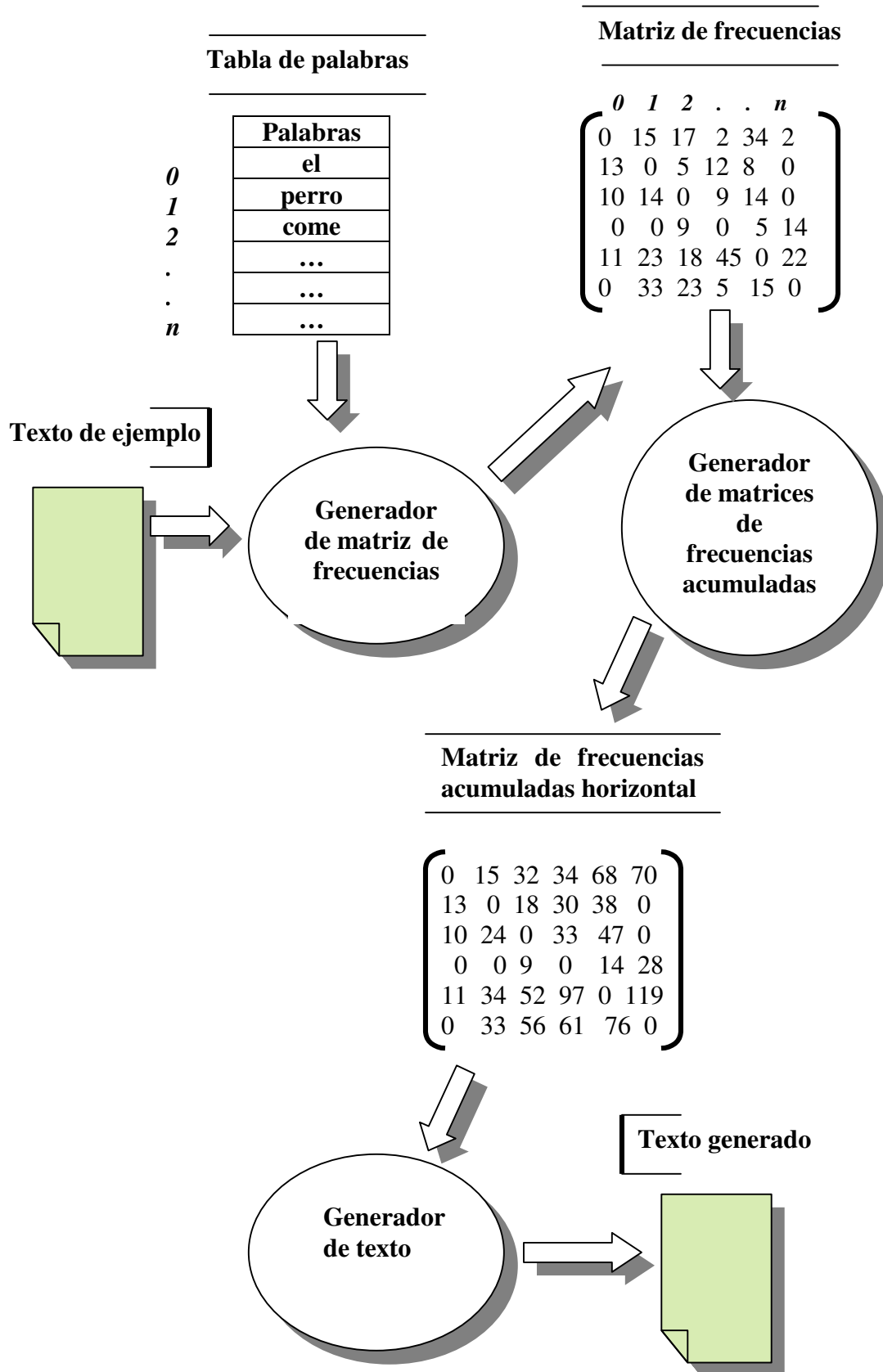


Fig.2.1 Manejador semántico

2.1 Generador de matriz de frecuencias

El generador de matrices de frecuencias tiene como objetivo encontrar la frecuencia con que una palabra sigue a otra dentro de un texto y construir la matriz que la representa. Genera una matriz de $n \times n$ donde n es el número total de palabras diferentes en el texto.

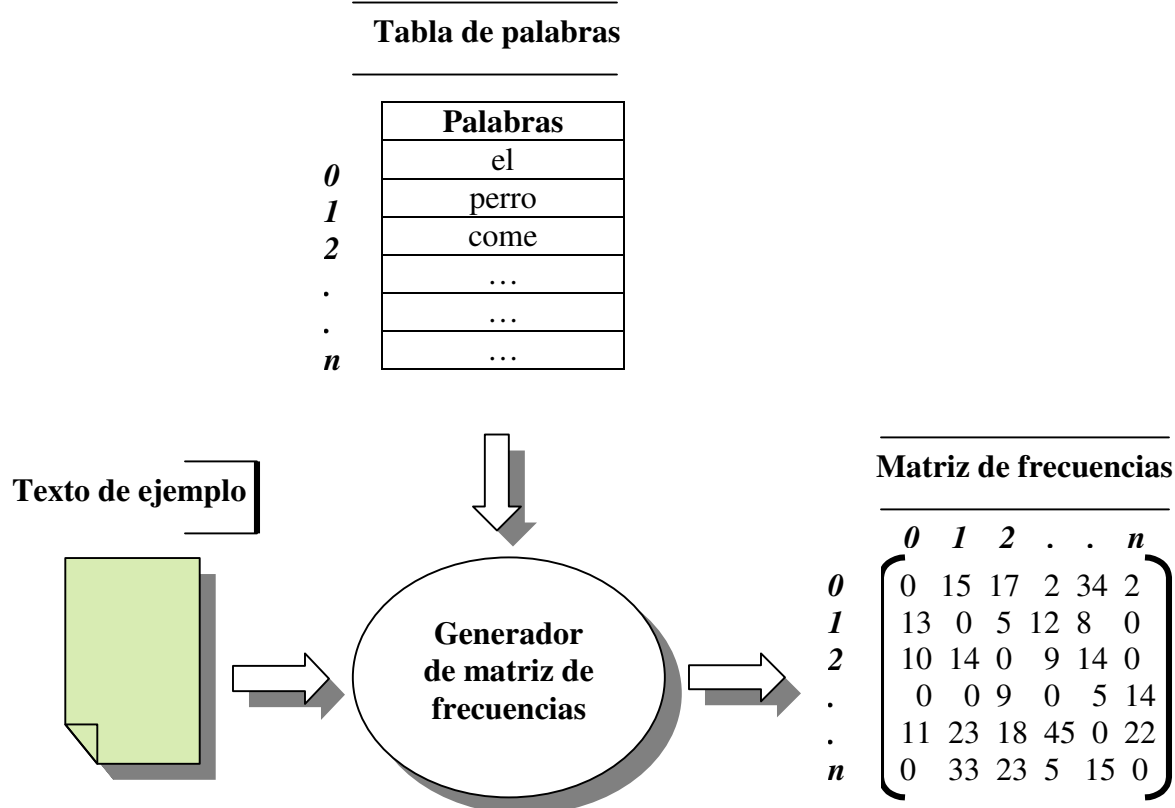


Fig. 2.2 Generador de matriz de frecuencias

El generador de matriz de frecuencias tiene como entrada un texto de ejemplo que en nuestro caso es del idioma español, pero que puede ser de otro idioma, como por ejemplo: *la niña juega en el patio con el gato y con la pelota.*

Lo que hace es tomar una palabra del texto, en este caso la palabra *la*, la localiza dentro de la tabla de palabras en donde tiene asignado un índice, en este caso 0.

Palabra1= la

Indice1= 0

Lee la siguiente palabra del texto *niña* y encuentra su índice, que equivale a 1.

Palabra2= niña

Indice1= 1

Enseguida incrementa el valor de la matriz de frecuencias en la posición (indice1, indice2), en este caso en la posición (0,1).

Por último, asigna la palabra2 a la palabra1 y busca la siguiente palabra en el texto, es decir, en este caso en particular, al principio

Palabra1 = la Índice1= 0
 Palabra2 = niña Índice2= 1
 Ahora
 Palabra1 = niña Índice1= 1
 y asignamos la siguiente palabra del texto a palabra2
 Palabra2 = juega Índice2= 3

Repite el proceso hasta que termina el texto de entrada, y obtenemos como resultado la matriz de frecuencias, que en este caso sería:

	la	niña	juega	en	el	patio	con	gato	y	pelota
la	0	1	0	0	0	0	0	0	0	1
niña	0	0	1	0	0	0	0	0	0	0
juega	0	0	0	1	0	0	0	0	0	0
En	0	0	0	0	1	0	0	0	0	0
El	0	0	0	0	0	1	0	1	0	0
patio	0	0	0	0	0	0	1	0	0	0
Con	1	0	0	0	1	0	0	0	0	0
gato	0	0	0	0	0	0	0	0	1	0
Y	0	0	0	0	0	0	1	0	0	0
pelota	0	0	0	0	0	0	0	0	0	0

2.2 Generador de Matrices de frecuencias acumuladas horizontal

Este generador tiene como objetivo generar la matriz de frecuencias acumuladas horizontal con la finalidad de poder generar texto hacia adelante. Y produce, al igual que el generador de la matriz de frecuencias, una matriz de $n \times n$. Tiene como entrada la matriz de frecuencias y lo que hace es acumular los valores de las casillas sobre el mismo renglón. Es importante aclarar que si alguno de los valores de la tabla vale 0 permanece como 0.

Lo que hace es lo siguiente:

Hasta llegar al renglón n

Mientras no termine el renglón, se van sumando los valores de las casillas con los valores de las casillas consecutivas y almacenando los valores en estas.

Por ejemplo, en la Fig.2.3 tenemos una matriz de frecuencias donde el primer renglón es:

(0 15 17 2 34 2)

Si acumulamos los valores el renglón de frecuencias acumuladas resultante es:

(0 15 32 34 68 70).
 (15+17) (32+2) (34+34) (68+2)

Este procedimiento es el mismo para cada uno de los renglones restantes.

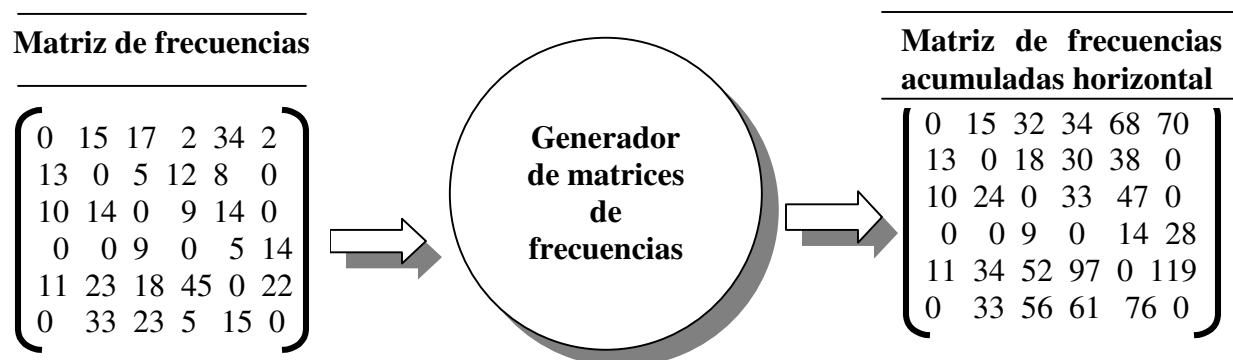


Fig.2.3 Generación de Matriz de frecuencias acumuladas horizontal

2.3 Generador de texto

El generador de textos, toma la matriz de frecuencias acumulada y la tabla de palabras y genera una secuencia de palabras o texto como se ve en la Fig.2.4.

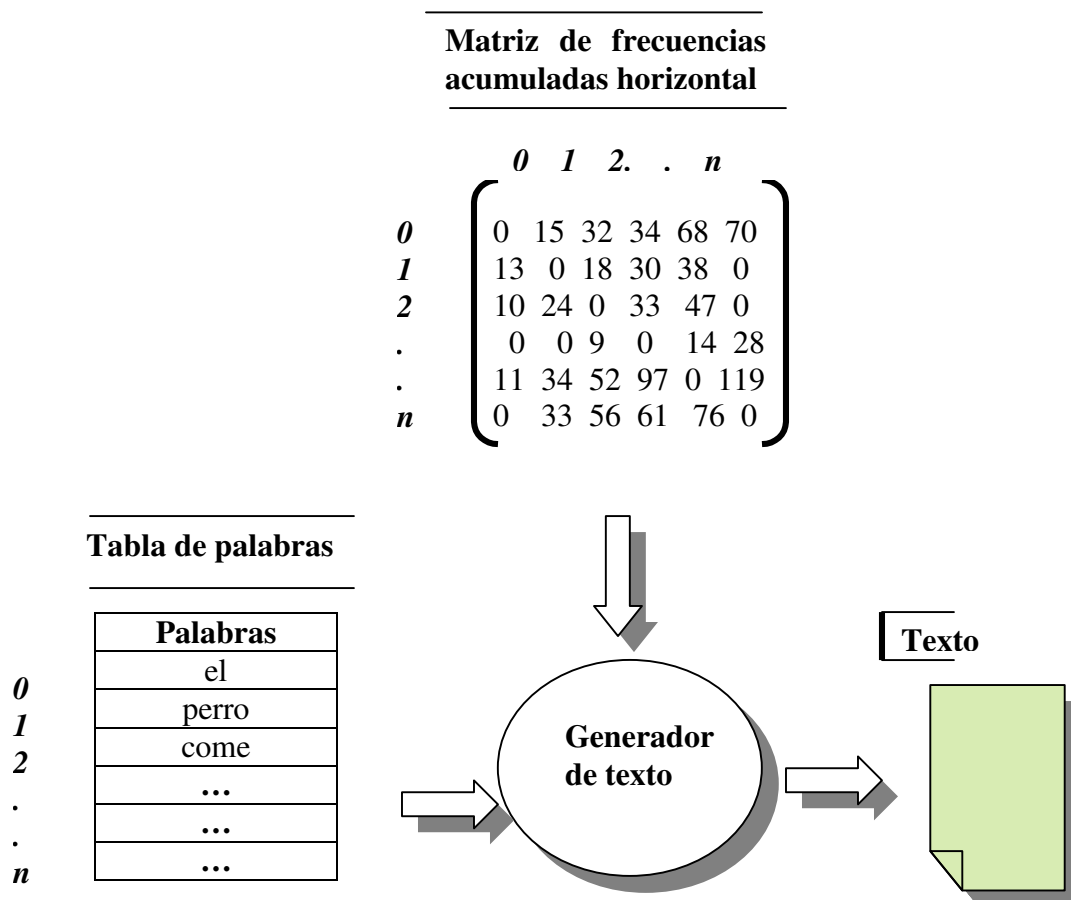


Fig. 2.4 Generador de texto

Lo que hace es lo siguiente:

Genera aleatoriamente un número

Toma dicho número como índice de la tabla de palabras

Genera la palabra a la que pertenece ese índice

Toma el renglón de la matriz de frecuencias acumuladas que corresponde al índice

Mientras no se cumpla la condición de terminación (Que puede ser por ejemplo que se encontró un punto '.' O que se genera cierta cantidad de palabras prefijadas).

Selecciona aleatoriamente algún valor diferente de 0 de ese renglón

ya que selecciono un valor

Toma el número de columna del nuevo valor seleccionado

Genera la palabra de la tabla de palabras que corresponde a ese número

Convierte el número de columna al número de renglón.

A continuación mostraremos como se pueden usar las matrices de frecuencias acumuladas para generar texto. Para lo cual partiremos del siguiente ejemplo

Para generar texto a partir de la matriz se selecciona aleatoriamente una palabra, en este caso '*perro*', y se posiciona en el renglón que corresponde a esta palabra y se comienza a generar texto a partir de ella.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro

Para lo cual se selecciona una de las columnas que sea diferente de 0 dentro del renglón '*perro*', y tomamos la palabra que corresponde a la columna seleccionada, en este caso '*está*'.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro

Se posiciona en el renglón que corresponde a la nueva palabra seleccionada, en este caso 'está' y se continua generando texto a partir de ella.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está

Para lo cual se selecciona una de las columnas que sea diferente de 0 dentro del renglón 'está' y buscamos la palabra que corresponde a la columna seleccionada, en este caso 'en'.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está

Se selecciona la palabra en este caso 'en' y se posiciona en el renglón que corresponde a esta palabra y se comienza a generar texto a partir de ella.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está en

Así sucesivamente hasta que se cumpla la condición de terminación (Que en este caso es cuando genera un punto '.' o cuando genera más de 10 palabras).

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está en

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está en el

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está en el jardín

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

perro está en el jardín .

2.4 Generador de Matrices de frecuencias acumuladas vertical

A partir del generador de textos ya se podrían ocultar palabras, ya que se puede primero generar un texto, luego inserta la palabra a ocultar y finalmente generar más texto. Pero al hacerlo nos dimos cuenta, que conviene que, tanto el texto que se genera antes de la palabra a ocultar, como el que lo sigue se generen a partir de la palabra a ocultar, por lo que se desarrollo también un mecanismo que genera texto a la izquierda de la palabra a ocultar, este mecanismo requiere de una matriz de frecuencias acumuladas vertical, por lo que en este punto se describe el generador de matriz de frecuencias acumuladas vertical.

Este generador tiene por objetivo generar una tabla de frecuencias acumuladas vertical, la cual sirve para ocultar el mensaje generando texto hacia atrás. Genera una matriz de $n \times n$ elementos.

Acumula los valores sobre la misma columna. Si alguno de los valores es 0 este no se modifica; lo que hace es lo siguiente:

Hasta llegar a la columna n

Mientras no termine la columna

Se van sumando los valores de las casillas consecutivas y almacenando los valores resultantes en estas.

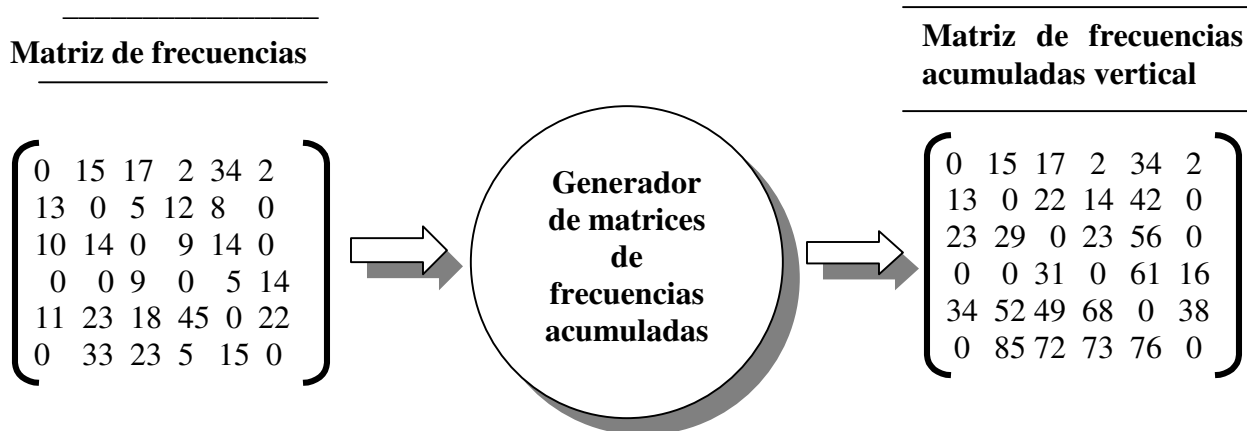


Fig.2.5 Generador de Matriz de frecuencias acumuladas vertical

Por ejemplo en la Fig.2.5 tenemos que la primera columna de la matriz de frecuencias es:

0
13
10
0
11
0

Por lo que la columna de frecuencias acumuladas vertical queda:

0	
13	0+13
23	13+10
0	No cambia
34	23+11
0	No cambia

Se sigue el mismo procedimiento para cada una de las columnas siguientes.

2.5 Generador de textos para ocultar mensajes

A continuación se presenta un algoritmo para el ocultamiento de las palabras del mensaje, el generador de texto para ocultar mensajes (Fig.2.6) tiene como entrada la tabla de frecuencias acumulada horizontal (para generar texto hacia delante) y la tabla de frecuencias acumuladas vertical (para generar texto hacia atrás), así como la tabla de palabras.

Lo que hace es lo siguiente:

Mientras no termine el mensaje a ocultar

 Toma la palabra a ocultar

 Toma el número de la palabra a ocultar como índice de la tabla de palabras

 Mientras no se cumpla la condición de terminación

 Toma el índice como columna de la matriz de frecuencias acumuladas vertical

 Selecciona aleatoriamente algún valor diferente de 0 de esa columna.

 Ya que selecciono un valor

 Toma el número de renglón del nuevo valor seleccionado

 Genera la palabra de la tabla de palabras que corresponde a ese número y las va colocando de derecha a izquierda al principio de la cadena generada (genera hacia atrás).

 Genera la palabra a ocultar

 Toma el número de la palabra a ocultar como índice de la Tabla de palabras

 Mientras no se cumpla la condición de terminación

 Toma el índice como renglón de la matriz de frecuencias acumuladas horizontal

 Selecciona aleatoriamente algún valor diferente de 0 de ese renglón.

 Ya que selecciono un valor

 Toma el número de columna del nuevo valor seleccionado

 Genera la palabra de la tabla de palabras que corresponde a ese índice y lo va colocando de izquierda a derecha, al final de la cadena generada (*genera hacia delante*)

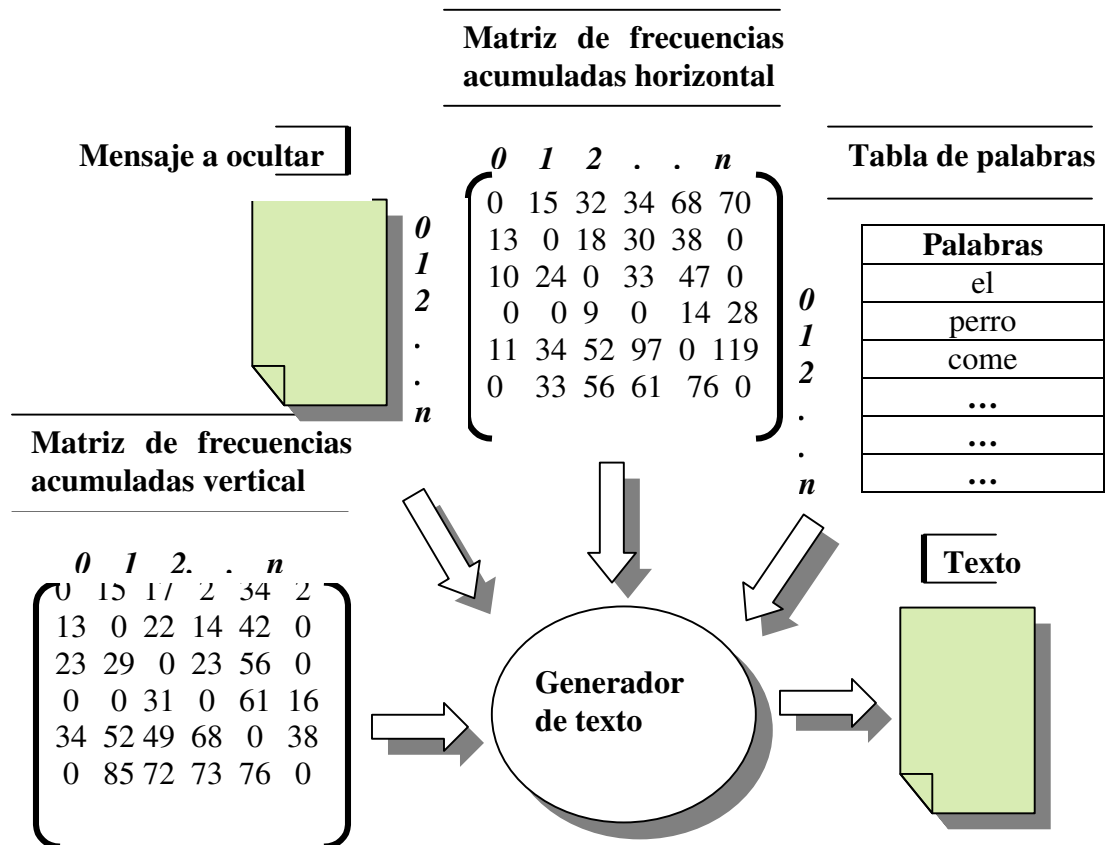


Fig.2.6 Generador de texto para ocultar mensaje

Por ejemplo, si la palabra que deseamos ocultar es 'está', localizamos la palabra 'está' como renglón en la matriz de frecuencias

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardin	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

está

Generamos texto hacia adelante a partir de la palabra 'está'

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

está en

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

está en la

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

está en la casa

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	3	0
perro	0	0	1	0	0	0	2	3	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	2	0	0	0	3	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	1	0	0	2	0	0
está	0	0	0	2	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	1	0	0	0	0	0	0	0
un	0	1	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	0	0	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	1	0	0
gato	0	0	0	0	0	0	1	0	0	0	0	0	0

está en la casa .

Ahora mostraremos la generación de texto hacia atrás a partir de la palabra 'está' con la ayuda de la matriz de frecuencias acumuladas vertical. Por principio, nos posicionamos en la columna que corresponde a la palabra 'está'.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	1	0
Perro	0	0	1	0	0	0	1	1	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	1	0	0	0	1	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	2	0	0	3	0	0
está	0	0	0	3	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	2	0	0	0	0	0	0	0
un	0	3	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	5	2	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	4	0	0
gato	0	0	0	0	0	0	3	0	0	0	0	0	0

está en la casa .

Se selecciona uno de los renglones que sea diferente de 0 dentro de la columna 'está', y buscamos la palabra que corresponde a la columna seleccionada, en este caso 'blanco'.

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	1	0
Perro	0	0	1	0	0	0	1	1	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	1	0	0	0	1	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	2	0	0	3	0	0
está	0	0	0	3	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	2	0	0	0	0	0	0	0
un	0	3	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	5	2	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	4	0	0
gato	0	0	0	0	0	0	3	0	0	0	0	0	0

está en la casa .

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	1	0
Perro	0	0	1	0	0	0	1	1	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	1	0	0	0	1	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	2	0	0	3	0	0
está	0	0	0	3	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	2	0	0	0	0	0	0	0
un	0	3	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	5	2	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	4	0	0
gato	0	0	0	0	0	0	3	0	0	0	0	0	0

blanco está en la casa .

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	1	0
Perro	0	0	1	0	0	0	1	1	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	1	0	0	0	1	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	2	0	0	3	0	0
está	0	0	0	3	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	2	0	0	0	0	0	0	0
un	0	3	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	5	2	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	4	0	0
gato	0	0	0	0	0	0	3	0	0	0	0	0	0

gato blanco está en la casa .

	el	perro	ladra	en	su	casa	blanco	está	la	un	.	jardin	gato
el	0	2	0	0	0	0	0	0	0	0	0	1	0
Perro	0	0	1	0	0	0	1	1	0	0	0	0	0
ladra	0	0	0	1	0	0	0	0	0	0	0	0	0
en	1	0	0	0	1	0	0	0	1	0	0	0	0
su	0	0	0	0	0	1	0	0	0	0	0	0	0
casa	0	0	0	0	0	0	0	0	0	0	2	0	0
blanco	0	0	0	0	0	0	0	2	0	0	3	0	0
está	0	0	0	3	0	0	0	0	0	0	0	0	0
la	0	0	0	0	0	2	0	0	0	0	0	0	0
un	0	3	0	0	0	0	0	0	0	0	0	0	2
.	2	0	0	0	0	5	2	0	0	3	0	0	0
jardín	0	0	0	0	0	0	0	0	0	0	4	0	0
gato	0	0	0	0	0	0	3	0	0	0	0	0	0

un gato blanco está en la casa .

Como se cumple la condición de fin (encontró un punto ‘.’), el sistema deja de generar texto hacia atrás, quedando oculto el mensaje.

CAPÍTULO 3 MÓDULOS DEL SISTEMA

3.1 Módulo Ocultador

Avanzando con el diseño de nuestro sistema llegamos a la integración de los elementos para construir el módulo ocultador.

Lo que pretendemos es obtener un programa que nos permita ocultar información por medio de una gramática libre de contexto para lo cual requerimos de ciertas características

3.1.1 Características

El mensaje debe tener la estructura según la secuencia de la gramática, es decir, si la secuencia de la gramática es que primero oculta un artículo, después oculta un sujeto y por ultimo oculta un verbo, la estructura del mensaje debe ser: artículo seguido de un sujeto, seguido de un verbo, ya que de otra manera en el texto con mensaje oculto se ocultaría por ejemplo, un verbo en lugar de un artículo o un sujeto en lugar de un verbo, lo cual ya no seguiría la gramática.

Cuando acaba el mensaje, termina de ocultar y acaba el algoritmo. Esta fue la solución más viable hasta el momento para que el extractor sepa hasta donde buscar el mensaje oculto.

3.1.2 Algoritmo

El algoritmo ocultador se basa en el Manejador de Gramáticas y el Generador de Textos. Como vemos en la Fig.3.1 al módulo ocultador entra un mensaje, el cual como ya se mencionó con anterioridad debe cumplir con la gramática, igualmente, tiene como entradas las tablas gramaticales y las matrices de frecuencias acumuladas.

El proceso que se lleva a cabo para ocultar el mensaje es el siguiente

Mientras no termine el mensaje

- Toma una palabra del mensaje
- Verifica que se encuentre en la tabla de unidades léxicas
- Obtiene el tipo al que pertenece dicha palabra
- Llama al Manejador de Gramáticas para el ocultador de textos

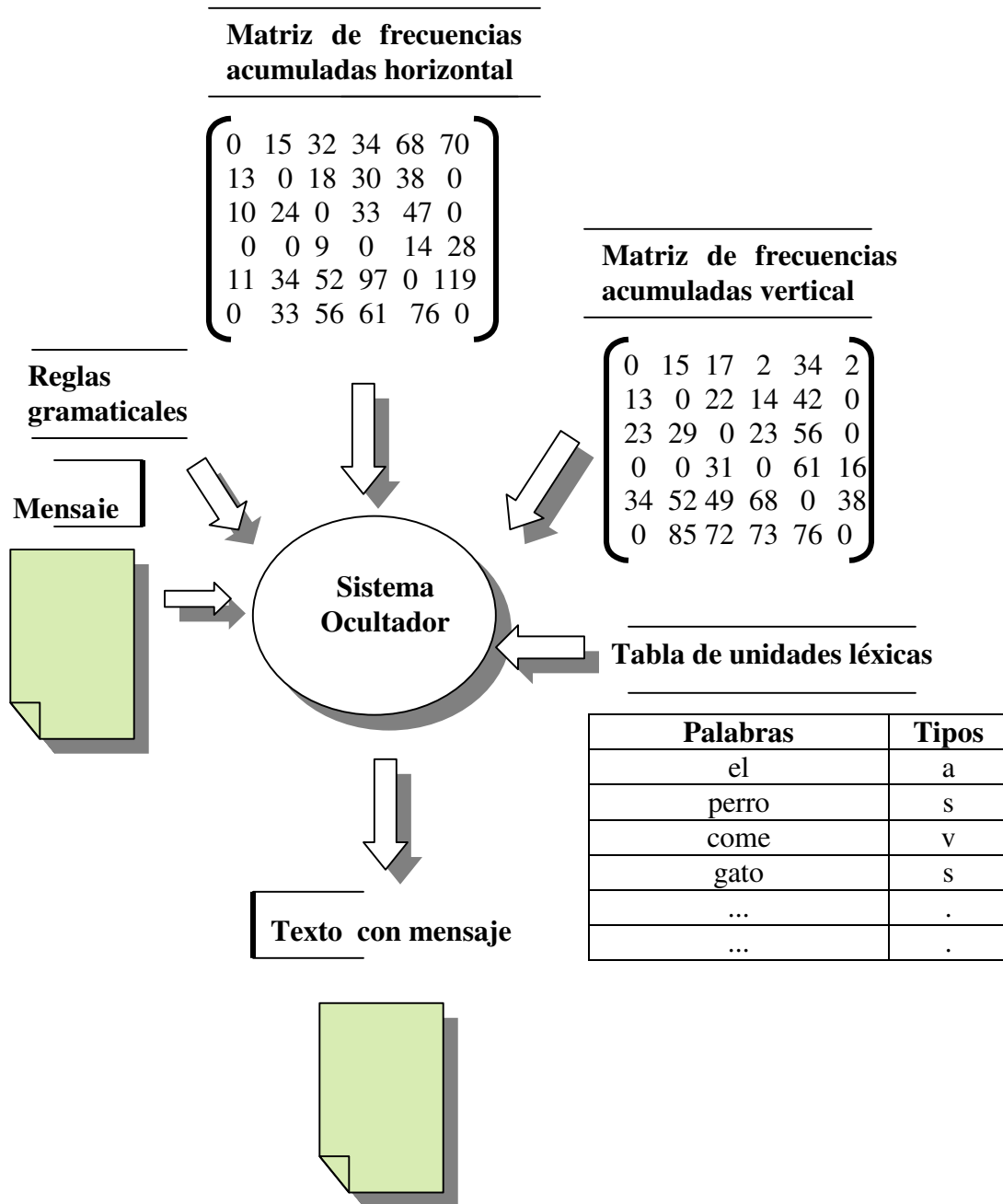


Fig. 3.1 Módulo Ocultador

3.1.3 Manejador de gramáticas para el ocultador de texto

El manejador de gramáticas para el ocultador se basa en el Manejador de Gramáticas presentado en la sección 1.2.1, la diferencia es que manda llamar al generador de textos ocultador, y su estructura se muestra en la fig. 3.2

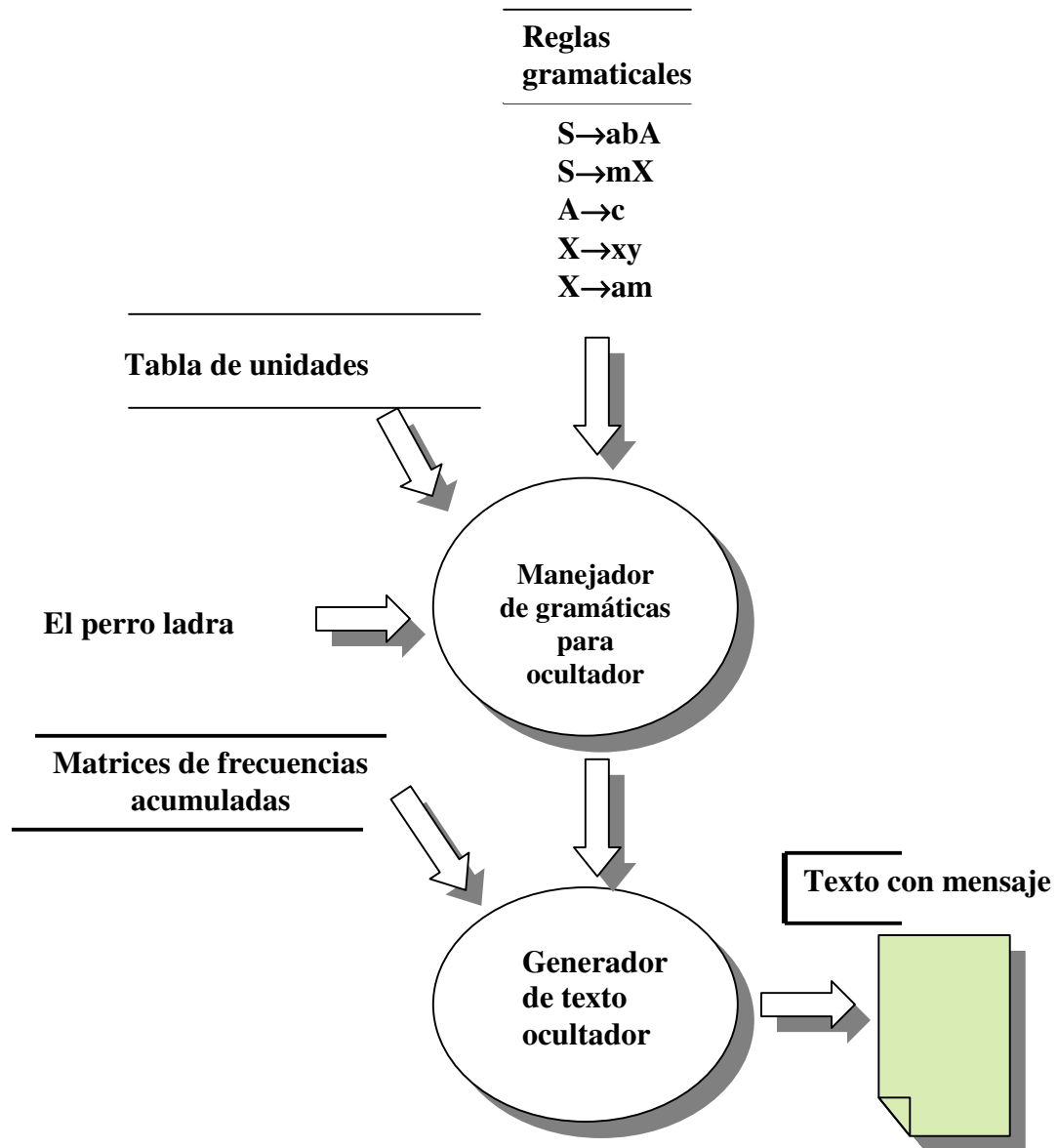


Fig. 3.2. Manejador de gramáticas para el ocultador

3.2 Módulo Extractor

Paralelamente al diseño del ocultador realizamos su contraparte que es un extractor que permite obtener el mensaje oculto.

3.2.1 Algoritmo

Como se puede observar en la Fig. 3.3 el Módulo Extractor tiene como entradas el texto con el mensaje oculto y las tablas gramaticales, las cuales son las mismas que las del Módulo Ocultador.

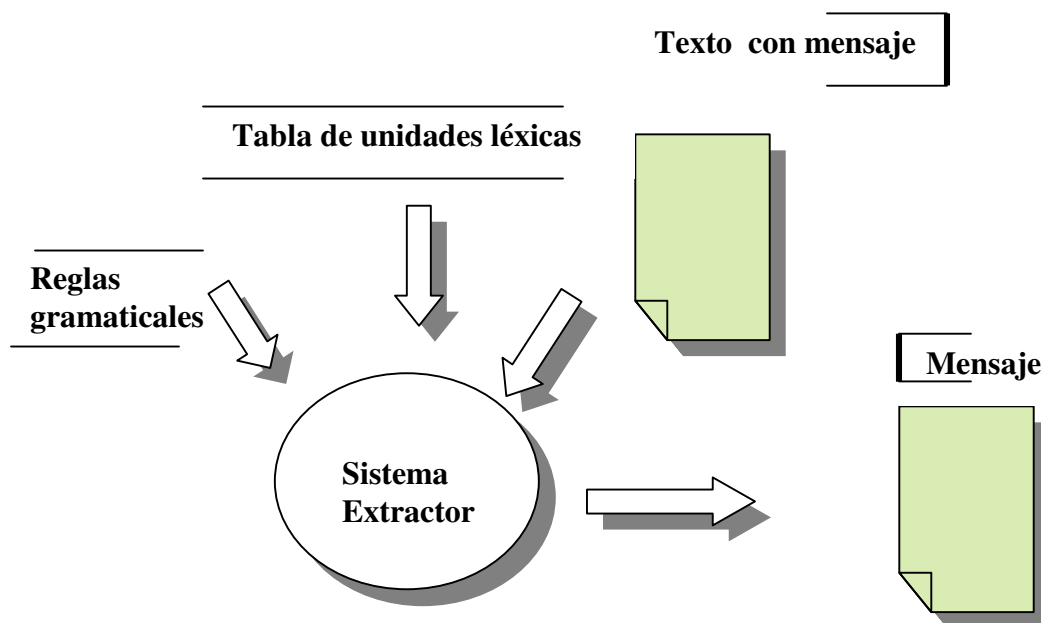


Fig. 3.3 Módulo Extractor

El proceso que se lleva a cabo para extraer el mensaje es el siguiente

Mientras no termine el texto con mensaje

- Toma una palabra del texto con mensaje

- Verifica que se encuentre en la tabla de unidades léxicas

- Obtiene el tipo al que pertenece dicha palabra

- Llama al Manejador de Gramáticas para el extractor de mensajes.

3.2.2 Manejador de gramáticas para el extractor de texto

El manejador de gramáticas para extraer un mensaje (Fig. 3.4) realiza lo siguiente:

Lee el texto con mensaje

Mientras no se termine el texto con mensaje

 Toma la siguiente palabra del texto

 Verifica que se encuentra en la tabla de unidades léxicas

 Obtiene el tipo al que pertenece

 Si corresponde con el tipo esperado según la gramática copia la palabra en el archivo de salida

 Si no corresponde, toma la siguiente palabra

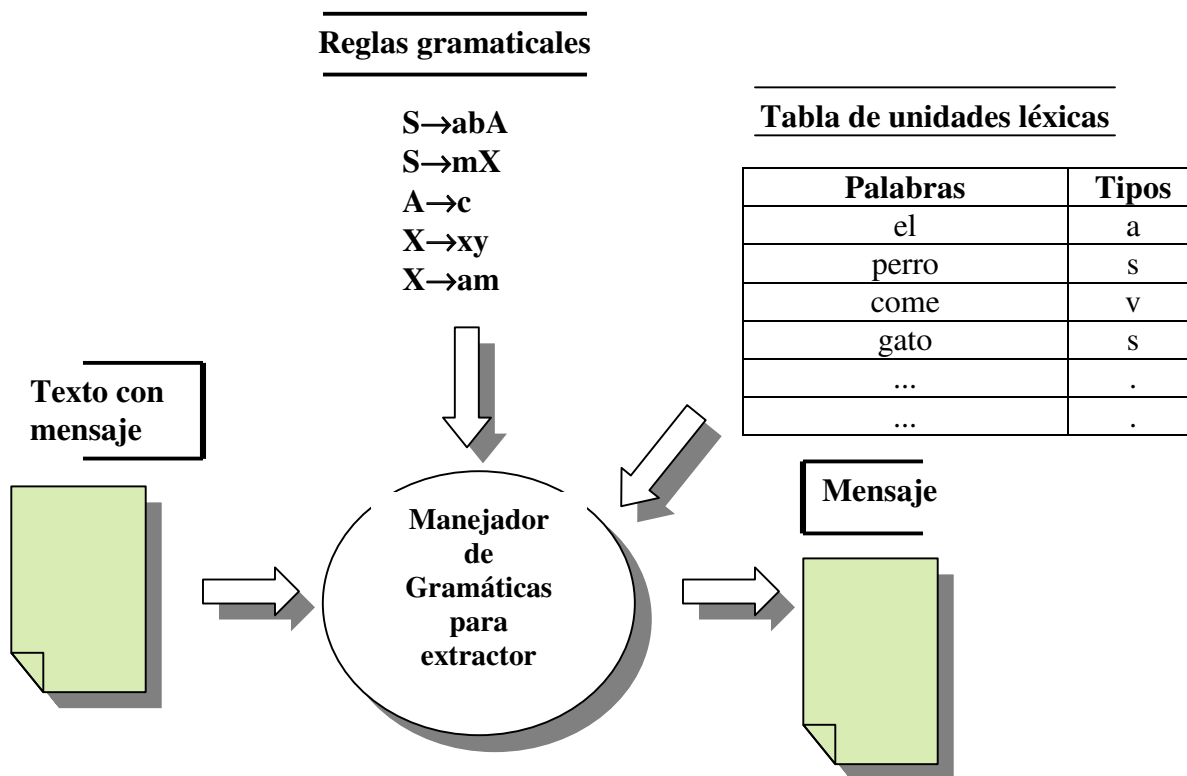


Fig. 3.4 Manejador de gramáticas para el extractor

Por ejemplo

Para extraer el mensaje que se encuentra en el siguiente texto

un gato blanco está en la casa.

Suponiendo que se tiene la siguiente regla gramática

$O \rightarrow v$

El manejador de gramáticas va verificando cada una de las palabras del texto con el mensaje oculto, si la palabra verificada es del tipo que indica la gramática, la extrae, de no ser así pasa a la siguiente palabra.

un gato blanco está en la casa .
p
un gato blanco está en la casa .
s
un gato blanco está en la casa .
c
un gato blanco está en la casa .
v

En este caso como la palabra *está* es la *primera palabra que aparece del tipo v*, el sistema la extrae.

El Módulo Extractor, extrae la primera palabra tipo *v*, por lo que el Módulo Ocultador debe verificar que el generador de texto no genere alguna palabra de tipo *v*, antes de la palabra que quiere ocultar.

En otro ejemplo, si se tiene la siguiente regla gramatical

$O \rightarrow p s v$

y el texto con mensaje oculto

a1 p a2 s a3 v a4
~~p~~ ~~s~~ ~~v~~

El sistema extrae la primera palabra de *tipo p*, luego la que sigue de *tipo s* y finalmente la que sigue de *tipo v*.

Por su lado el Módulo Ocultador debe verificar que no se inserte una palabra de *tipo p* antes de la primera palabra a ocultar, ni una de *tipo s* entre la primera y la segunda o una de *tipo v* entre la segunda y la tercera.

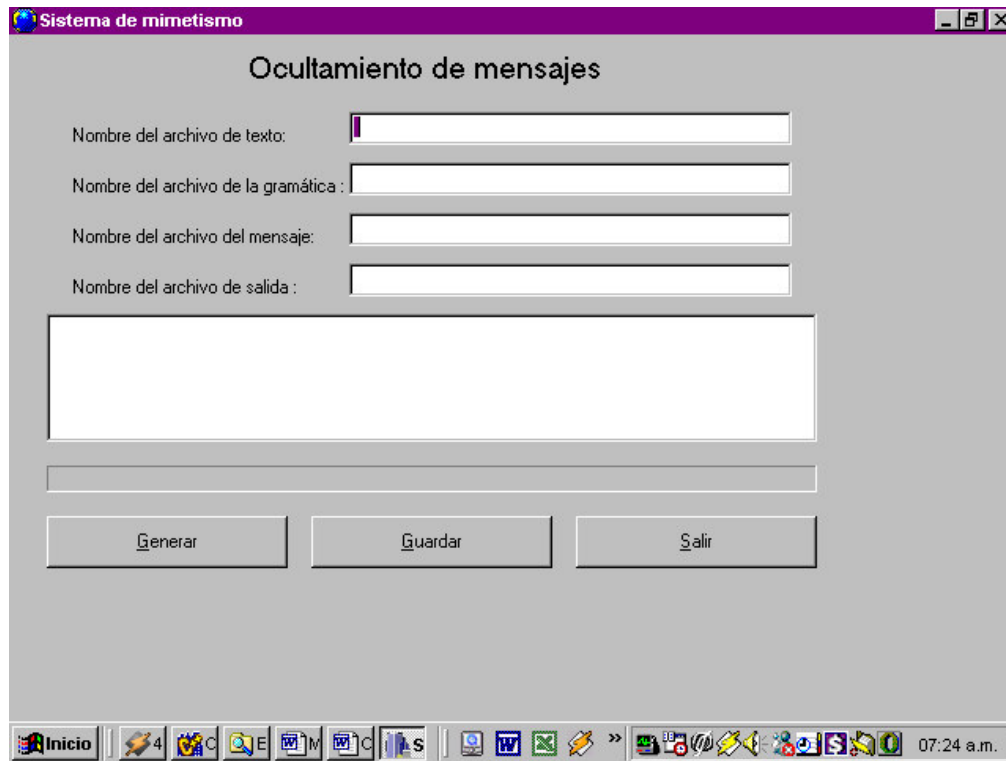
Con lo cual se ve que, *la relación que se da entre el Módulo Ocultador y el Módulo Extractor es mediante la gramática* ya que el modulo ocultador oculta el mensaje de acuerdo a la gramática y el extractor extrae siguiendo *la misma* gramática.

También se observa la relación entre el generador de texto y el manejador de gramáticas que se da en el Módulo Ocultador y *que la Gramática es la llave que permite ocultar y recuperar mensajes en este sistema.*

CAPITULO 4 RESULTADOS.

A continuación se muestran ejemplos demostrativos del funcionamiento del sistema.

Cuando llamamos al sistema de ocultamiento nos pone la siguiente pantalla.



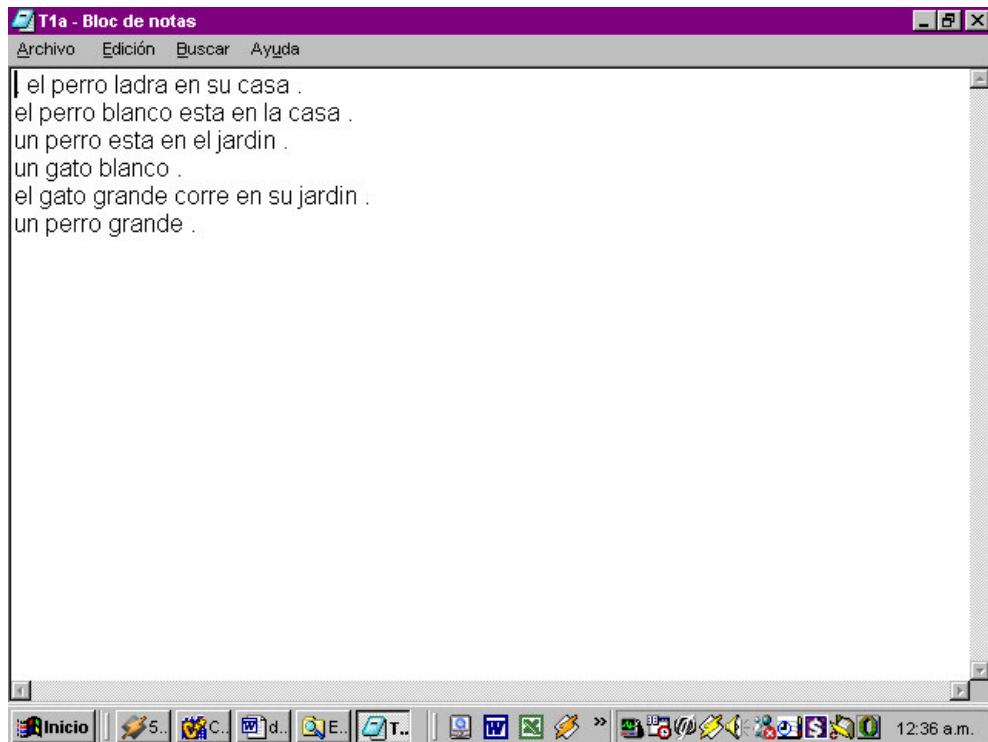
Donde nos piden el nombre de archivo de textos de ejemplo, el nombre de archivo de la gramática, donde se introducen las reglas gramaticales a usar, la lista de elementos terminales, la tabla de unidades léxicas y el nombre del archivo donde esta el mensaje a ocultar.

Por lo que a continuación veremos varios ejemplos, donde se mostrará el contenido de los archivos de entrada, como se proporcionan al sistema y los resultados obtenidos.

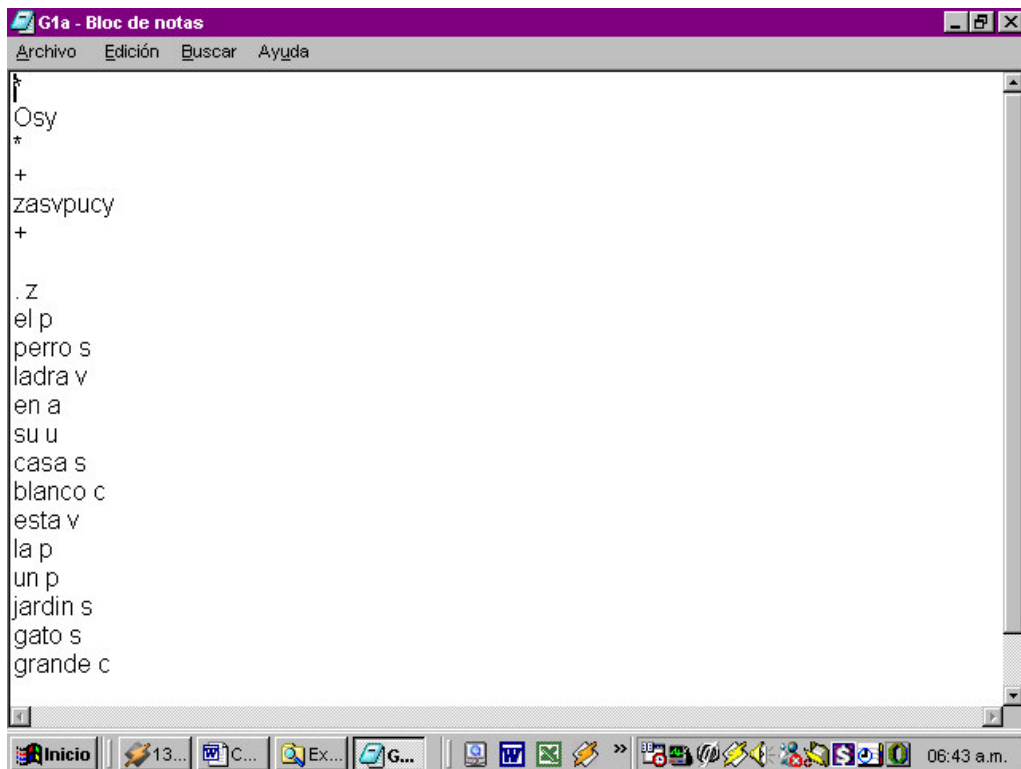
4.1 Ejemplo 1.

Primero para mostrar el funcionamiento del sistema veremos un ejemplo sencillo, ya que, solo oculta una palabra.

A continuación se muestra el archivo de texto *T1a.txt*, el cual contiene los *textos de ejemplo* que se introducirán al sistema.



El archivo de la gramática, que se guardo en *G1.txt* es el siguiente:

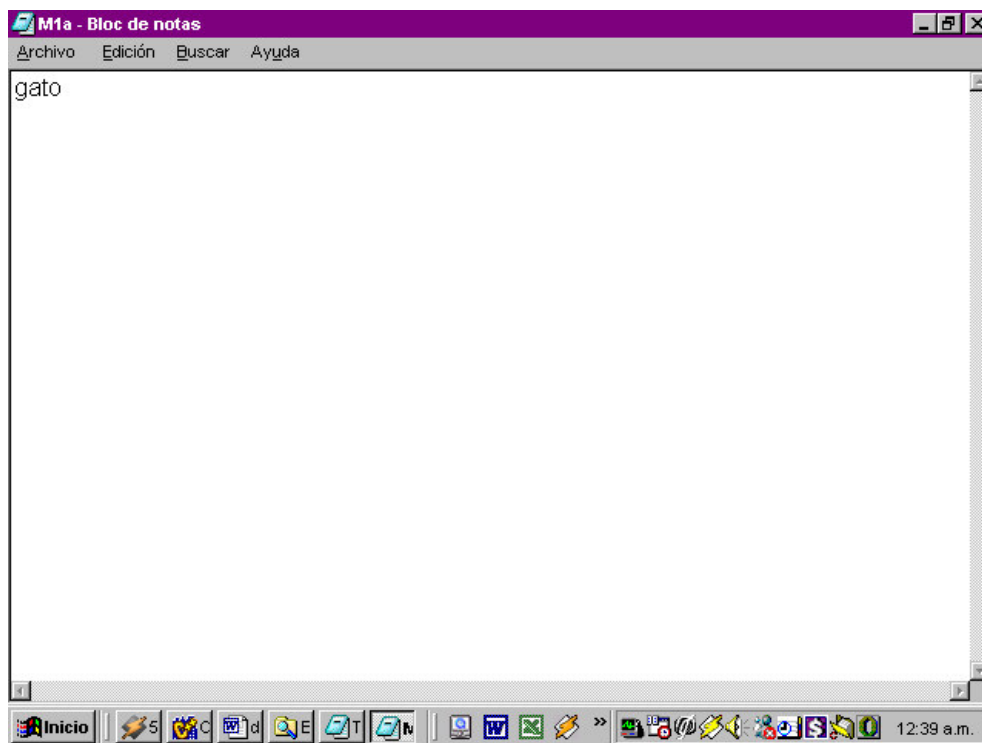


En este archivo primero introducimos la regla de producción, donde indicamos que la oración *O* se compone de un sujeto *s*, y el símbolo *y* que implementa nuestro sistema como símbolo terminador ($O \rightarrow sy$).

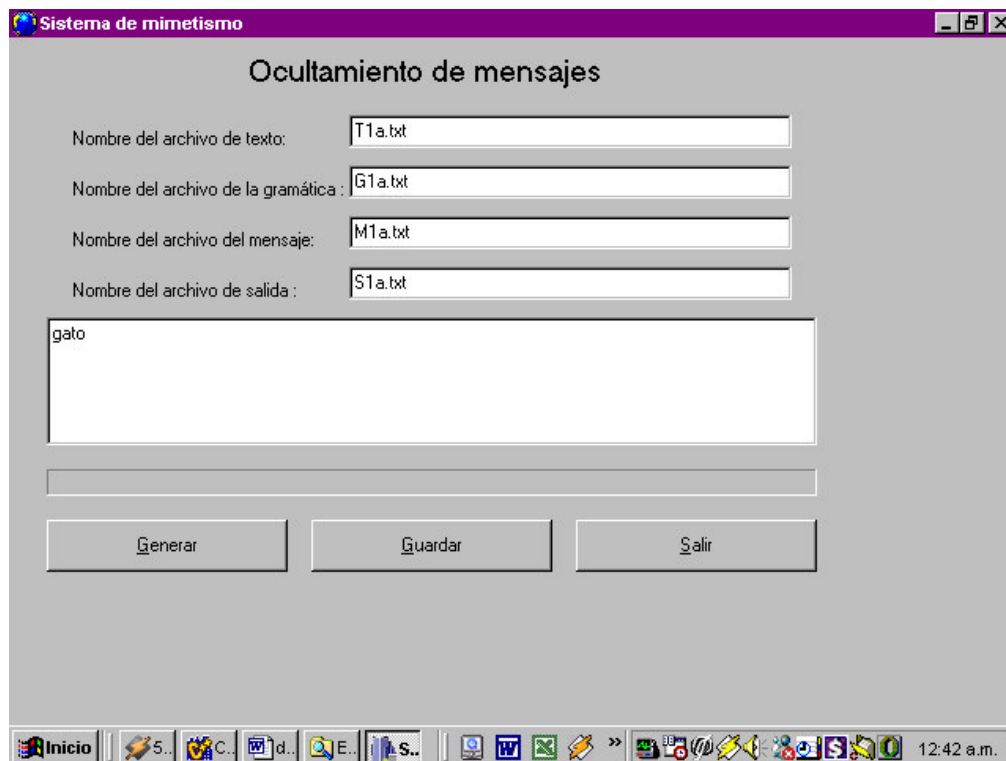
Luego se incluye la lista de los tipos de elementos terminales (*c*, *p*, *a*, *d*, *s*, *m*, *n*, *v*, *x*, *y*), donde el ultimo elemento indicado es el símbolo terminador (*y*).

Además en este archivo se incluye la tabla de unidades léxicas, donde se tienen las palabras que acepta el sistema y sus tipos.

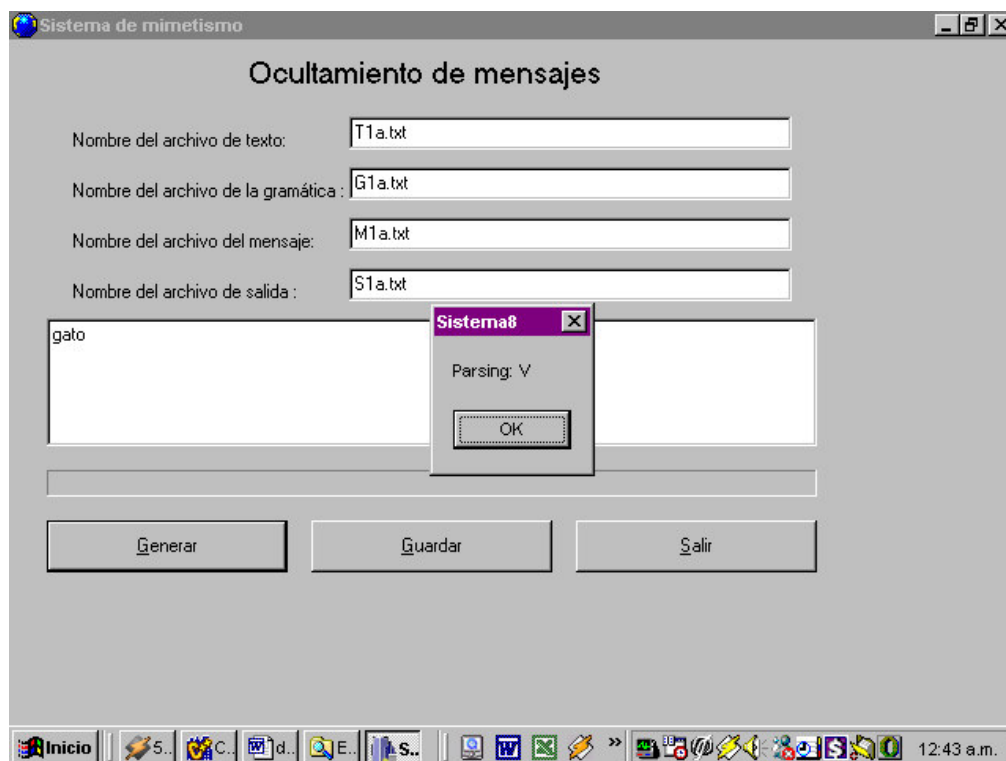
Ahora, mostramos el *archivo del mensaje a ocultar M1a.txt* el cual se corresponde con la estructura definida por la gramática (la cual, como ya se mencionó para este caso se compone de un sujeto).



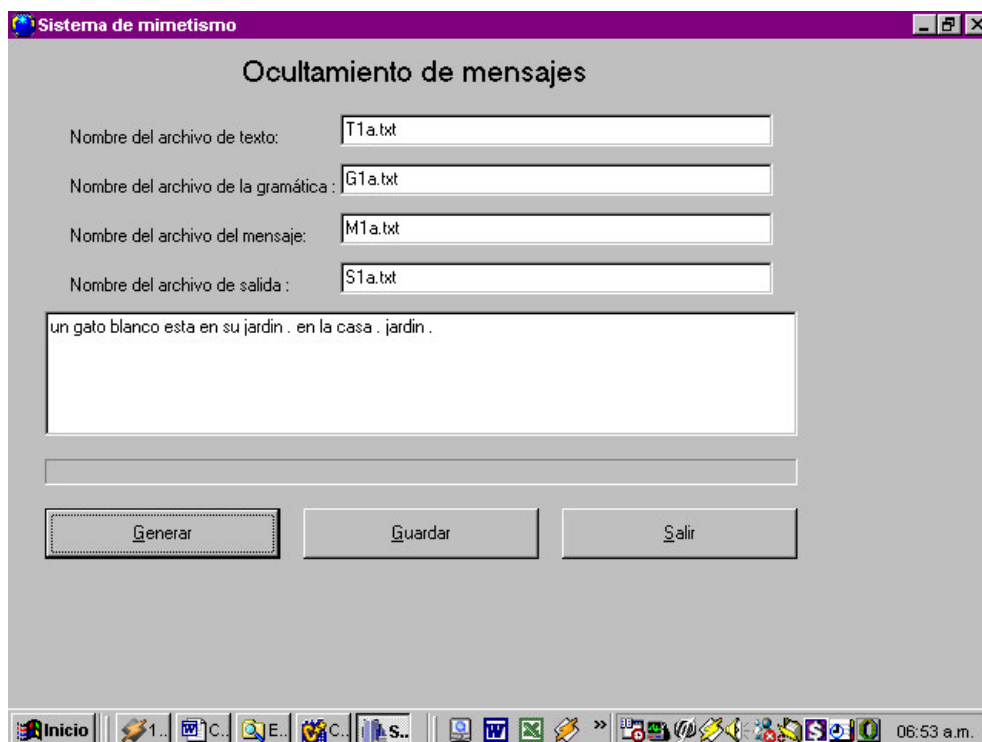
En la *pantalla de Ocultamiento* de nuestro sistema le indicamos los archivos de entrada ya mostrados, el nombre del *archivo con textos de ejemplo T1a.txt*, el nombre del *archivo con la gramática G1a.txt*, el nombre del *archivo con el mensaje M1a.txt*, y el nombre del *archivo de salida, S1a.txt*. Esto se muestra en la siguiente imagen, en la que se observa el *mensaje a ocultar en la caja de texto, este es el sujeto gato*.



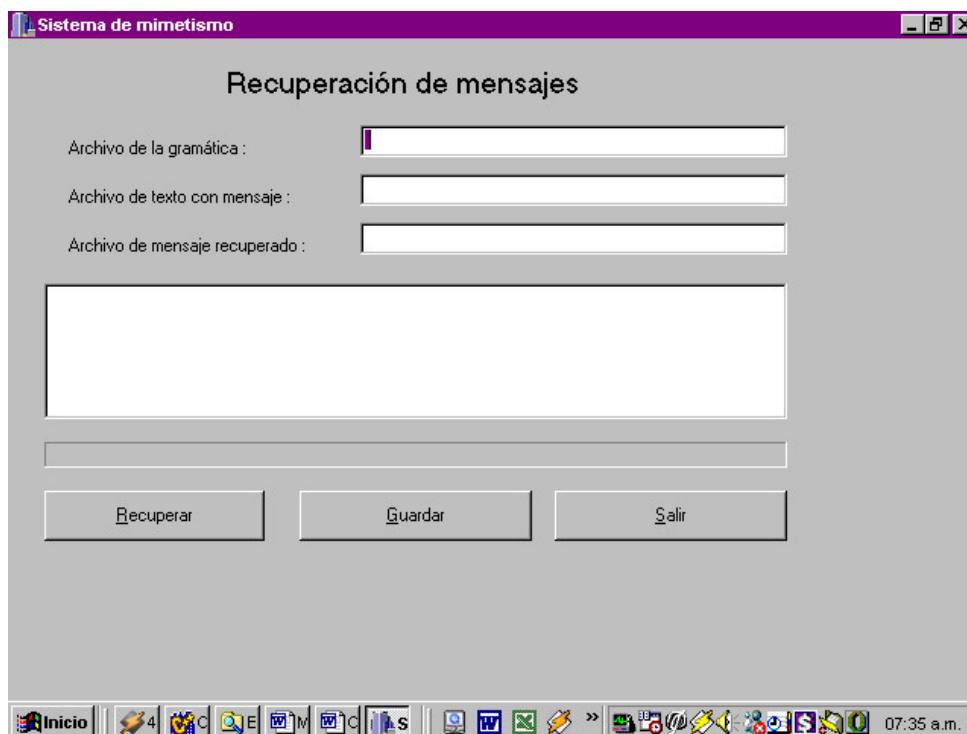
Al seleccionar la opción *generar pantalla de Ocultamiento de mensajes*, nos mostrara el resultado del análisis, por medio de una caja de mensaje, como se muestra a continuación.



Al teclear en el *botón OK*, observaremos el texto con el mensaje oculto generado en la siguiente pantalla.

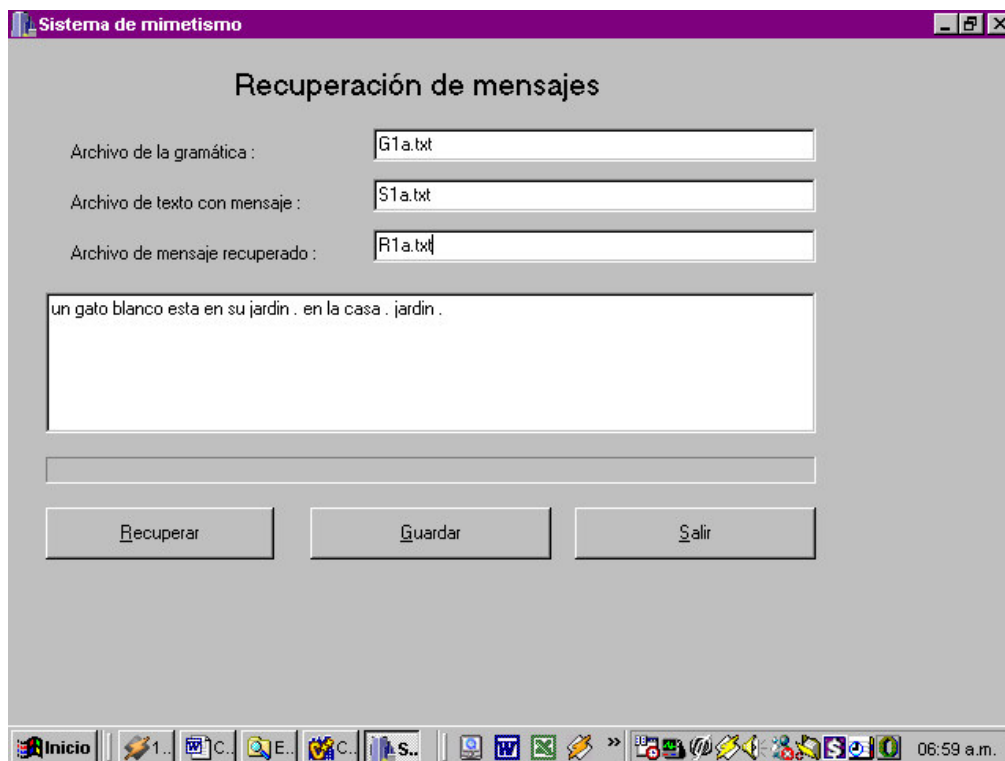


Para recuperar el mensaje, tenemos que usar el *sistema de recuperación de mensaje*, el cual tiene la siguiente pantalla.

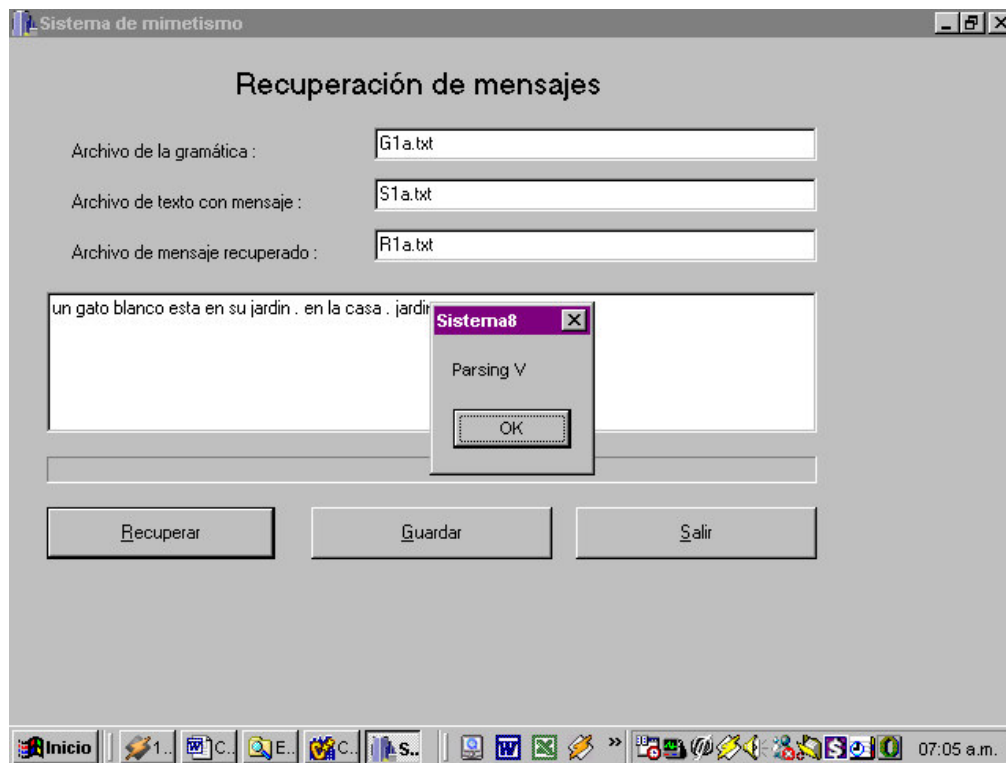


Como se observa en esta pantalla, se solicitan los nombres de los archivos donde esta la gramática, el texto con mensaje generado en el sistema ocultador y el nombre del archivo donde se almacenara el mensaje recuperado.

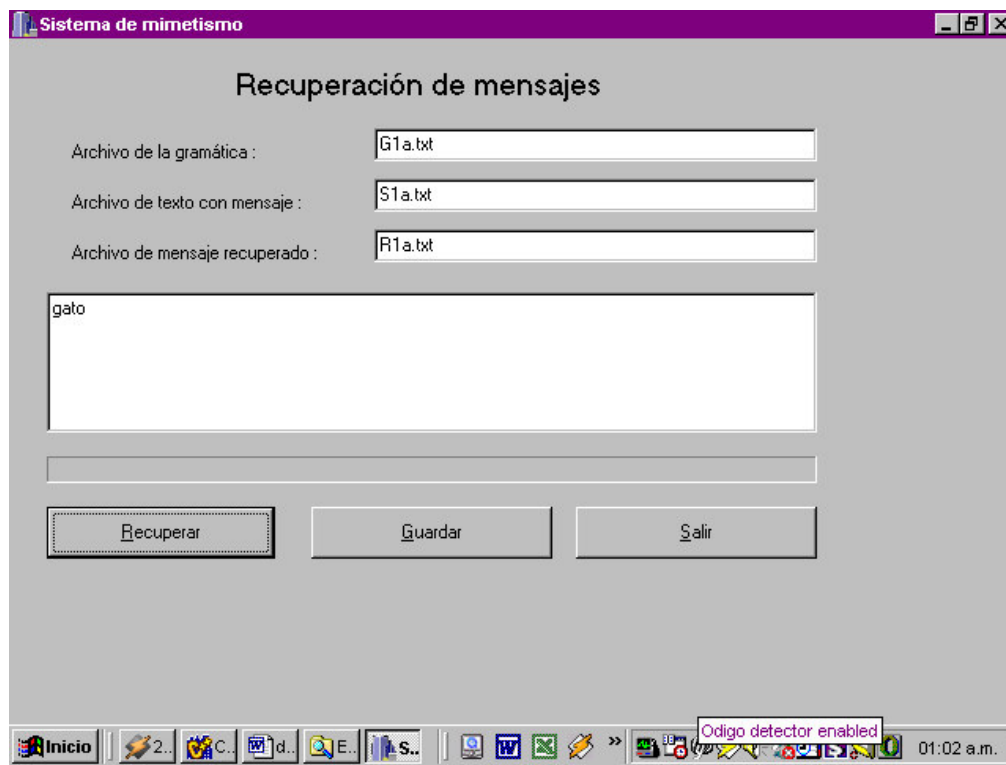
Entonces en la pantalla para *la recuperación del mensaje*, se indican los nombres de los archivo de las entradas, G1a.txt que tiene la gramática, S1a.txt que tiene el texto con el mensaje oculto generado por el proceso anterior y el nombre del archivo R1a.txt, que contendrá el mensaje recuperado.



Cuando seleccionamos la opción Recuperar en la pantalla de *recuperación de mensajes*, el sistema nos entrega el resultado del proceso de recuperación por medio de una caja de mensajes, donde aparece V o F dependiendo si pudo o no recuperar el mensaje, como se muestra a continuación.

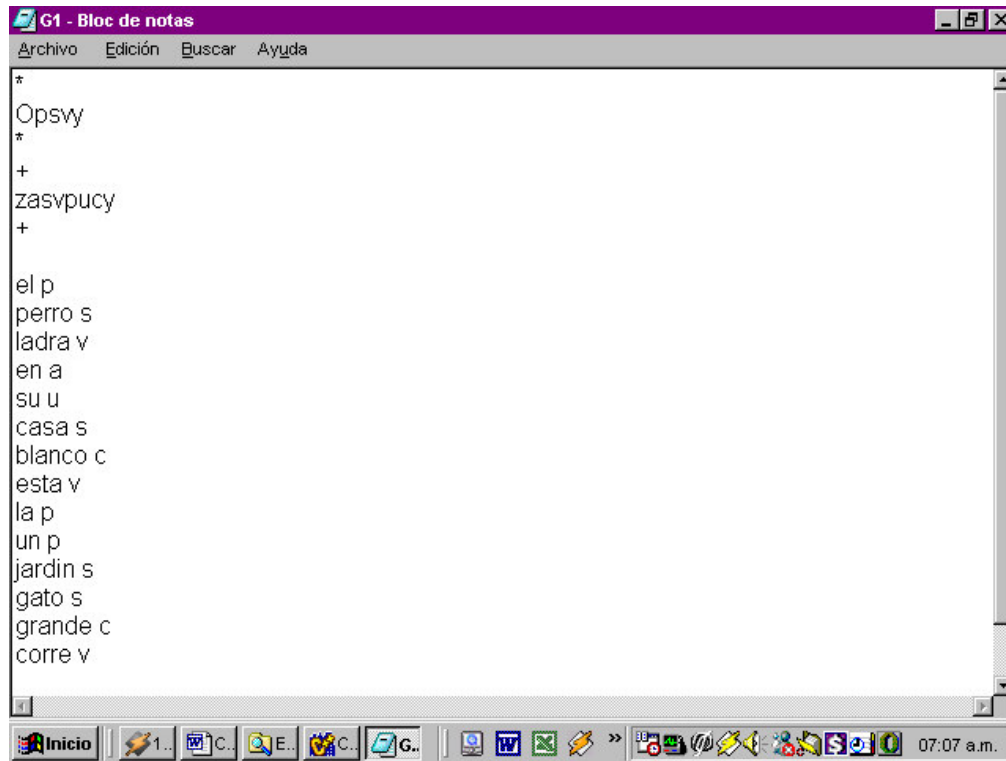


Al teclear en el *botón OK*, observaremos el mensaje oculto recuperado en la siguiente pantalla, el cual es efectivamente el mensaje original, el sujeto *gato*.



4.2 Ejemplo 2.

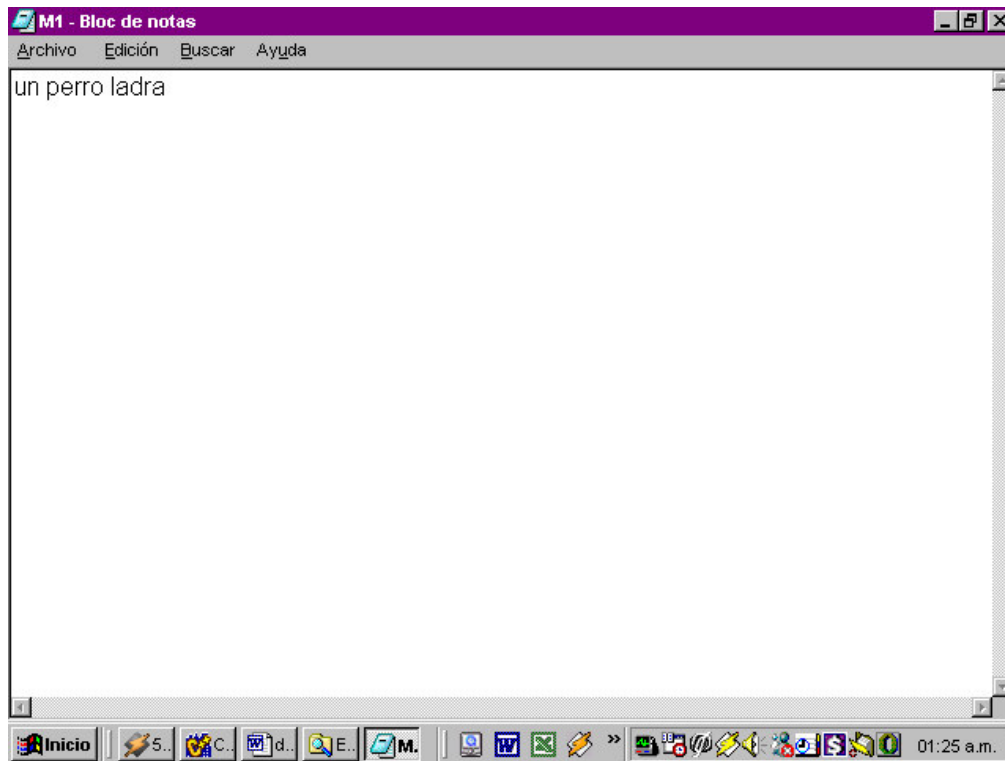
El siguiente ejemplo se ejecuta con una variación en la estructura de la gramática esta será la siguiente.



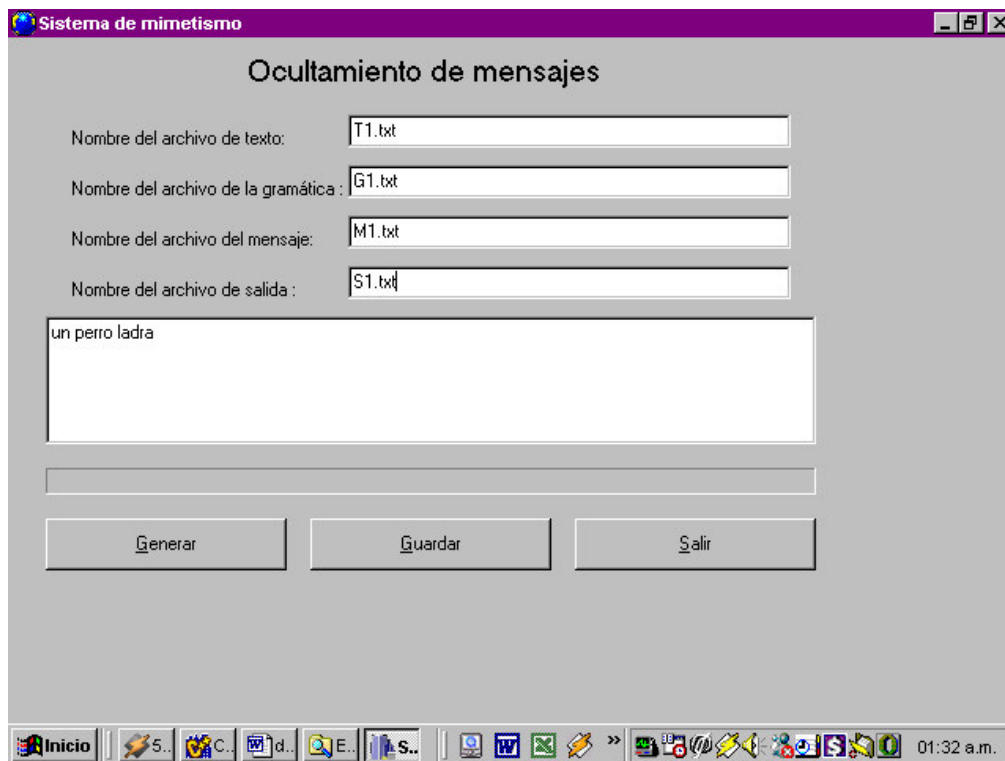
Esto es, la oración **O** se compone de un pronombre **p** seguido de un sujeto **s** y un verbo **v** además de el símbolo **y** que implementa nuestro sistema como símbolo terminador ($O \rightarrow \text{psvy}$).

La tabla de unidades léxicas y el texto de ejemplo son los mismos del ejemplo 1.

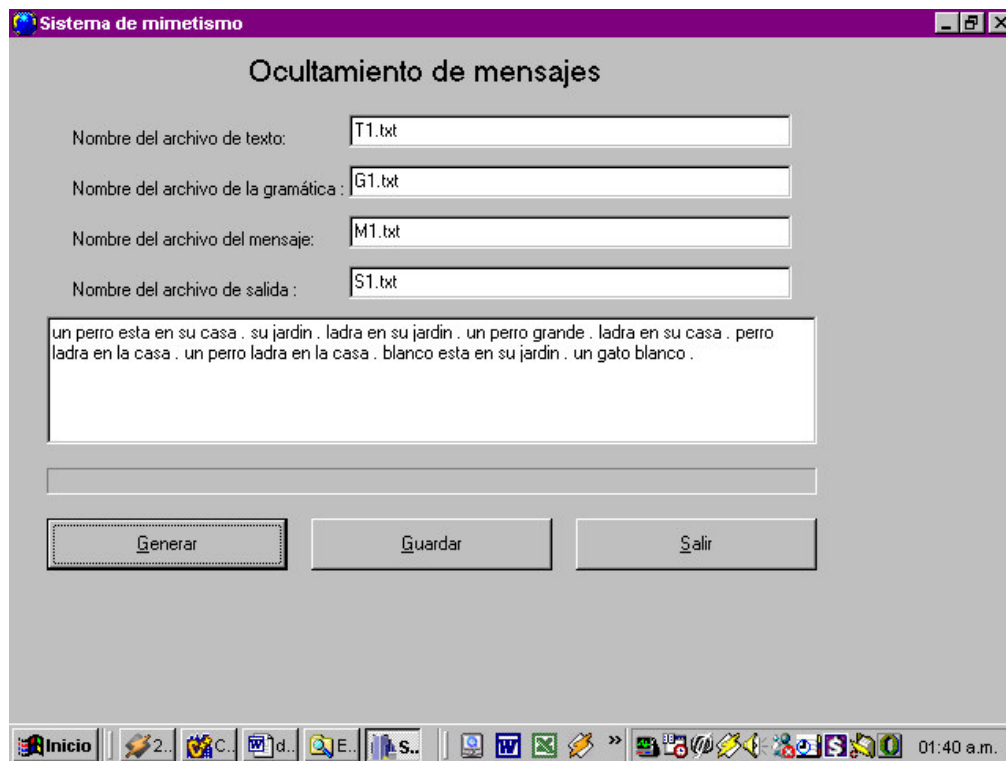
El mensaje a ocultar es el siguiente.



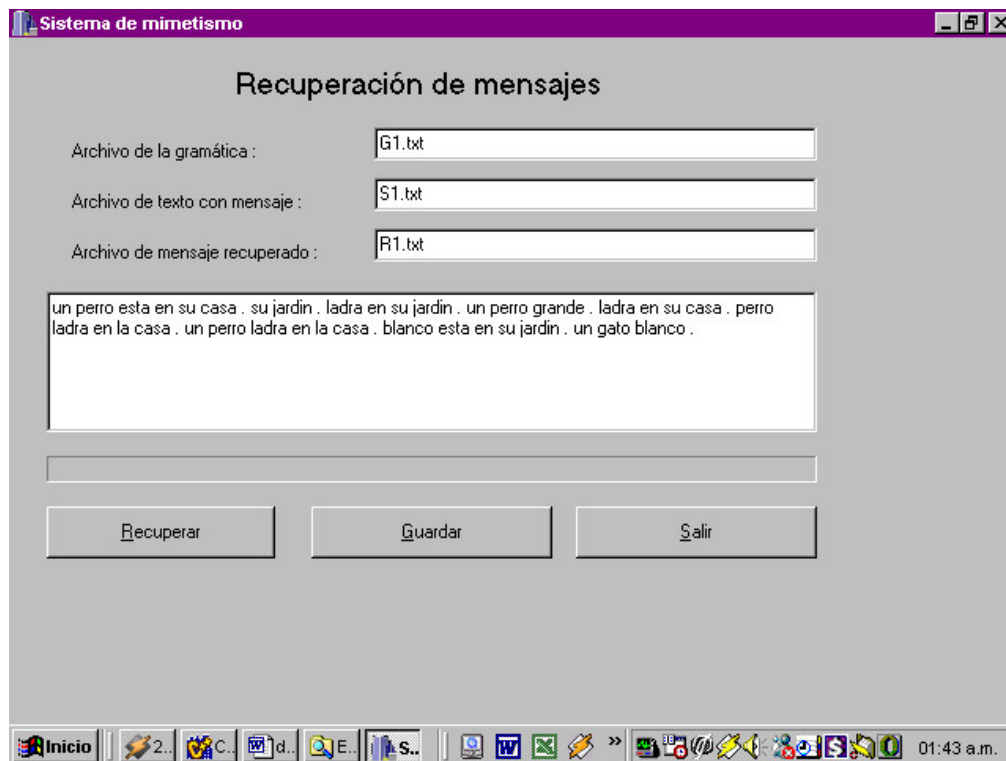
Estos componentes los indicamos en la pantalla de ocultamiento de mensajes del sistema , como se observa a continuación.



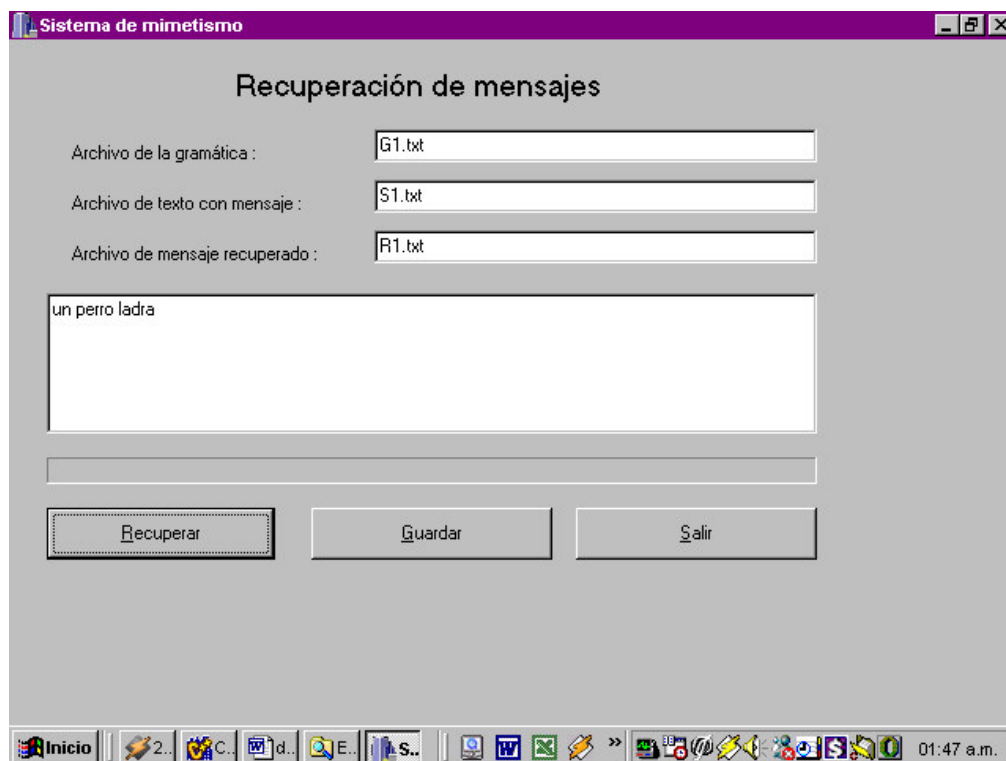
Y en la siguiente pantalla mostramos el texto con mensaje, generado por el ocultador de mensajes.



Posteriormente este archivo se recibe en el proceso de recuperación de mensajes. Se indica que se desea almacenar en el archivo R1.txt como se muestra en la siguiente pantalla.



Y al recuperar el mensaje, aparecerá en el cuadro de texto, el cual como observamos a continuación es el mensaje original *un perro ladra*.



4.3 Ejemplo 3.

El siguiente ejemplo cuenta con la siguiente estructura gramatical

*

OAx

ApAc

As

*

+

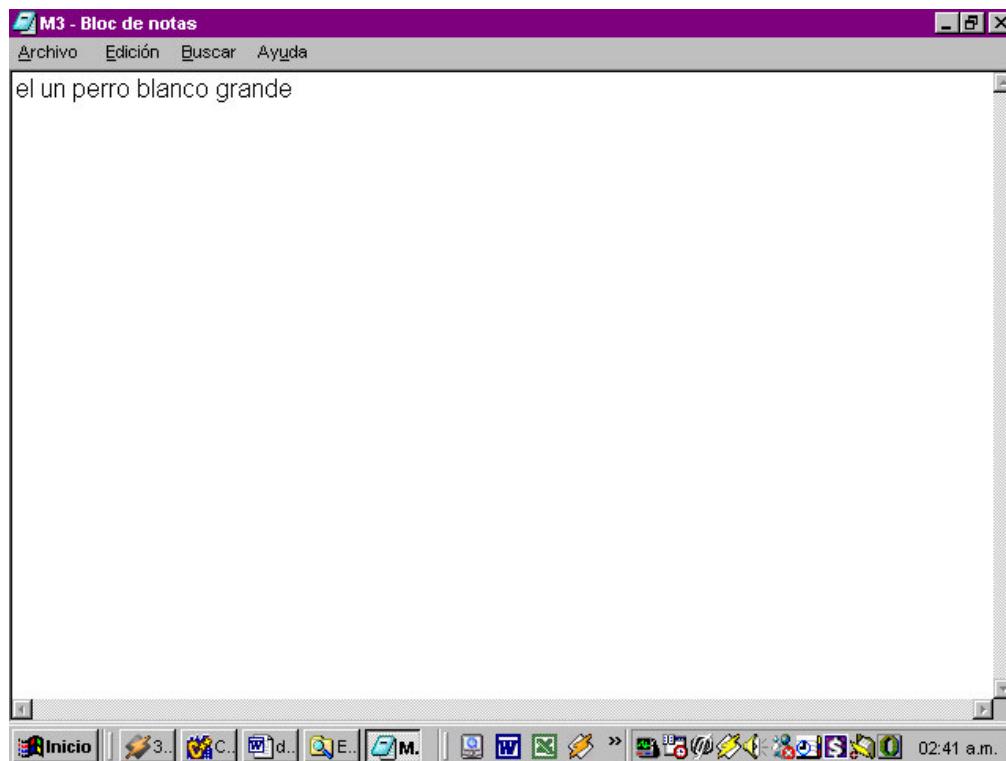
zasvpucx

+

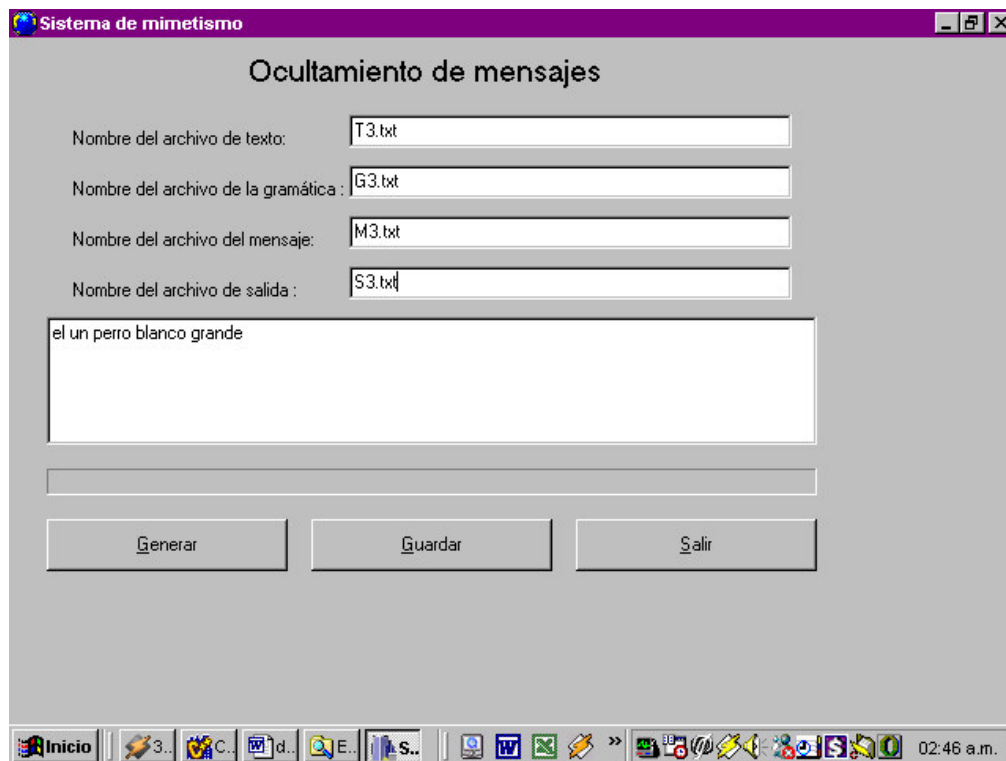
Que como se ve *es una gramática libre de contexto con recursividad empotrada*. La tabla y el texto implementados son los mismos. El mensaje a ocultar es

el un perro blanco grande

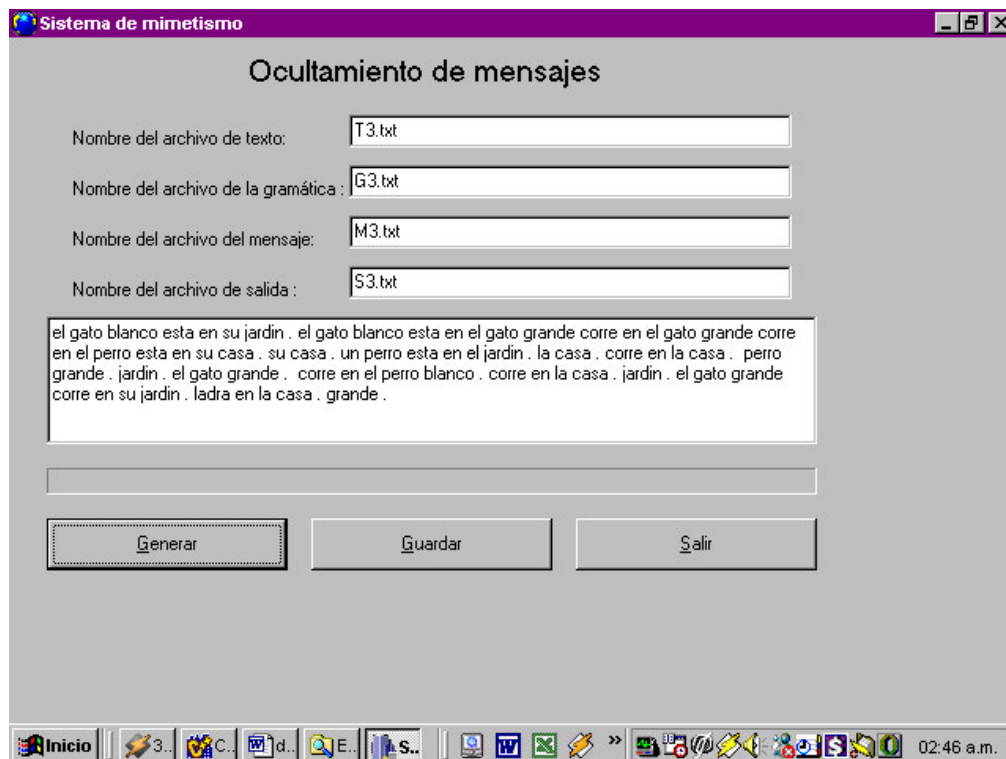
y a continuación se muestra.



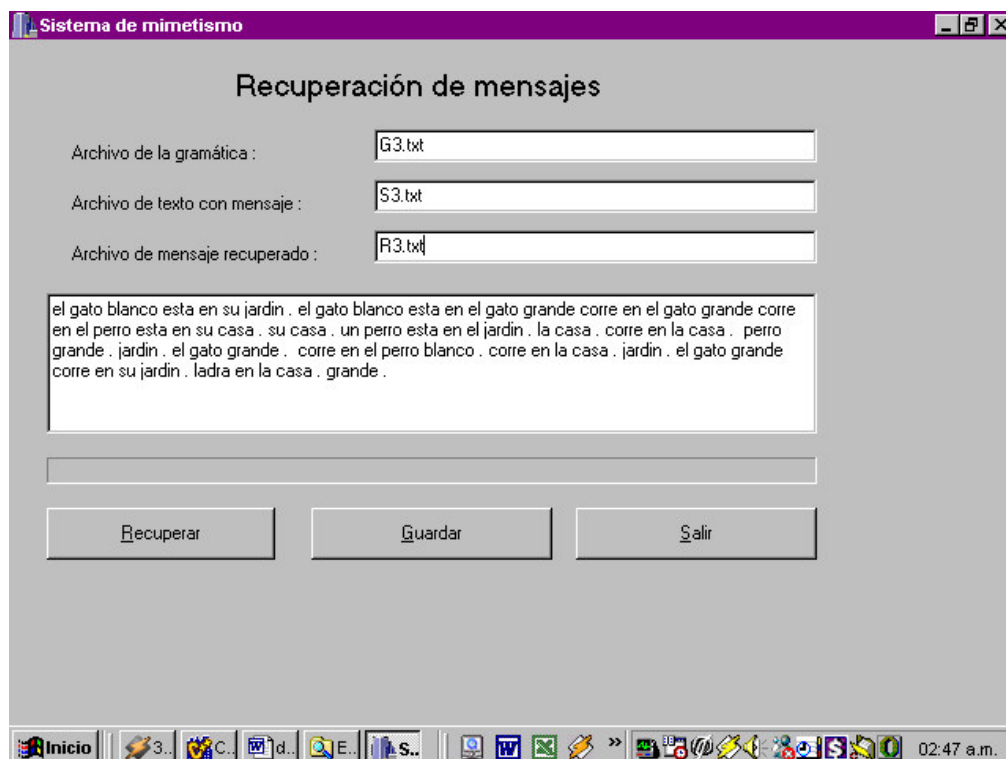
Le indicamos los elementos en la pantalla de ocultamiento, en este caso el *archivo con textos de ejemplo* esta en *T3.txt*, el *archivo con la gramática* en *G3.txt*, el nombre del *archivo del mensaje a ocultar* en *M3.txt* y el *archivo de salida con el texto con mensaje oculto* es *S3.txt*.



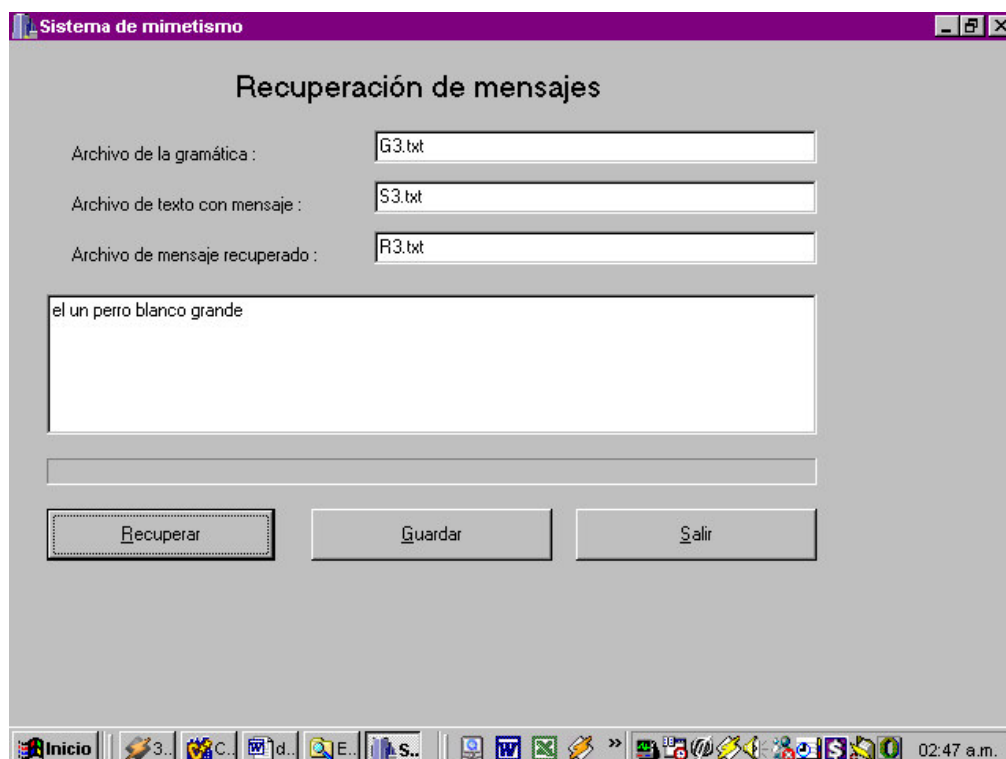
Al ejecutarse el sistema ocultador nos generará el texto con mensaje, como se muestra en la figura siguiente.



Ahora proporcionamos al sistema de recuperación de mensajes, el archivo de la gramática G3.txt, el archivo S3.txt, de texto con mensaje, e indicamos el archivo R3.txt como destino.



Posteriormente elegimos Recuperar, para que obtenga el mensaje oculto.



Y como se observa es *el un perro blanco grande*, el mensaje original

4.4 Ejemplo 4.

Ahora, presentaremos otro ejemplo con variaciones en los textos de entrada, en la gramática y por consiguiente en el mensaje a ocultar.

Su gramática es la siguiente.

Gramática.

*
OIFx
IpaNdN
IaNdN
NsL
Lm
Lnm
FpaN
FaN
*
+
pasmnx
+

La tabla de unidades léxicas es la siguiente.

en p	la a	ESCOM s	nos r
titulamos v	con c	un n	Proyecto s
. z	el a	consiste e	una n
investigacion s	evalua v	comision s	esta e
formada v	por n	profesores s	los a
tienen v	conocimiento n	del d	area x
Veces s	son n	muy m	estrictos x
profesor s	es v	bueno x	su u
puede v	ser n	malo x	alumnos s
se u	graduan v	calificacion x	selecciona v
diversos x	trabajos s	claros m	entienden v
mejor m	las a	anteriores n	oraciones s
conforman v	serie x	de d	enunciados s
que x	han n	elaborado v	para x
Describir v	poco x	funcionamiento n	respecto x
a p	sus a	trabajos s	finales m
explicaremos v	mas m	sobre n	esto x

ultimo m	año s	desarrollan v	trabajo s
final m	, z	este p	y c
desarrollo v	especifica m	debe v	desarrollar n
equipos s	lo a	menos n	dos m
estudiantes s	organizan v	presentan v	propuestas s
semestre s	antes n	documentos s	describe v
propone v	calendario n	actividades m	metodologia s
vez x	entregadas v	esperan x	aceptacion v
realizarlo v	si n	rechazan v	algunas x
buscar v	algun x	otro n	no m
siempre x	suelen n	rechazar v	algunos n
casos m	hacen v	correcciones x	recomendaciones n
tomen m	cuenta x	propuesta s	rechazada v
porque n	criterio m	cumple v	suficiente x
considerado n	como m	titulacion x	depende v
gran n	medida m	evaluacion v	proyectos a
estos x	terminales m	vera v	deben n
calidad m	tiene v	seguir n	alguna m
forma x	podra n	revisar v	periodicamente m
avances x	siguiendo v	pues x	asigna v
parte n	resto m	asignan v	director s
evaluadores m	ya x	descripcion n	clara m
cual r	influyen v	tanto n	administrativos s
presentaciones s	entregan v	documento s	previo m
acerca n	explicaciones m	claras m	puesto n
texto s	sirbe v	base n	puedan n
realizar v	correcta m	dia x	presentacion s
textos s	eficaces m	presentar v	detalladamente n
toda m	empleada v	resultados n	obtenidos v
conclusiones s	objetivo s	tenga v	sistema s
desarrollado v	tambien n	poder n	evaluarlo v
rapidamente m	aun r	sean v	o c
manos n	metodos m	trabajan v	manera n
adecuada m	oportuna m	claro x	despues n
preparan v	donde n	evaluan v	exponen v
ante r	siguieron n	fueron v	ventajas n
desventajas n	estas x	metodologias s	posibles n
mejoras m	paso n	proceso s	programa v
cada n	equipo s	asignaran v	porcentaje x
alumno s	apoyar v	cualquier x	herramienta s
exposicion v	le d	sea x	util v
realizan v	periodo n	semana m	hay v
cierta m	gente s	que x	no x
conoce v	el a	funcionamiento n	

El archivo de *textos de ejemplo T4.txt*, contiene lo siguiente:

. hay cierta gente que no conoce el funcionamiento de la ESCOM , para describir a grandes rasgos este , se ha elaborado un documento con la colaboracion de algunos alumnos de la ESCOM .

estos alumnos mencionan el tipo de problemas que se presentan en este funcionamiento , especificamente en los ultimos semestres , con los trabajos terminales . en muy grandes rasgos se mencionan lo siguiente .

en la ESCOM nos titulamos con un proyecto .

el tipo de los proyectos son diversos .

otros proyectos no lo son .

el proyecto consiste en una investigacion .

la investigacion la evalua la comision .

la comision esta formada por profesores .

los profesores tienen conocimiento del area .

los temas del area son diversos .

los profesores aveces son muy estrictos .

un profesor es muy bueno en su area . otros no .

un profesor puede ser muy malo calificando .

los alumnos se graduan con su calificacion .

la comision selecciona diversos profesores .

los trabajos claros se entienden mejor .

pues estos trabajos ya son finales .

los trabajos con propuestas son documentos .

las anteriores oraciones conforman una serie de enunciados , que se han elaborado para describir un poco el funcionamiento en la ESCOM con respecto a sus trabajos finales .

asi , explicaremos un poco mas sobre esto . respecto a otros temas se hablara mas adelante , acerca de esto se desarrollo un documento . este tipo de documentos son para la comision . el tipo de este es de trabajos finales .

los alumnos de ESCOM en el ultimo año desarrollan un trabajo final , este trabajo es de investigacion y desarrollo en un area especifica . el trabajo final se debe desarrollar en equipos de por lo menos dos estudiantes , los estudiantes se organizan y presentan sus propuestas un semestre antes .

ademas , cuando organizan sus equipos , los alumnos deben tener , ademas un asesor el cual los guiara y tambien evaluara su trabajo . la colaboracion del asesor es importante para el desarrollo del trabajo , esta colaboracion es a veces tecnica , otras veces es tactica y en otros casos es hasta moral .

por eso es importante buscar un buen asesor para que los pueda guiar periodicamente y su colaboracion sea clara . cuando un equipo le hace la propuesta a algun profesor para ser su asesor , estos esperan su respuesta para para que los alumnos lo tomen en cuenta .

las propuestas de proyectos son documentos en los que se describe el proyecto , se propone un calendario de actividades y una metodologia de desarrollo . una vez entregadas las propuestas , los alumnos esperan la aceptacion de su trabajo para realizarlo .

si se rechazan algunas propuestas los alumnos tienen que buscar algun otro proyecto . no siempre se suelen rechazar propuestas , en algunos casos se hacen correcciones y recomendaciones para que los alumnos las tomen en cuenta .

una propuesta se puede rechazar porque a criterio del profesor no cumple con lo suficiente para ser considerado como proyecto de titulacion . cuando se rechazan propuestas si se tienen que presentar otras propuestas .

la titulacion de los alumnos depende en gran medida de el desarrollo y evaluacion de sus proyectos . estos son trabajos terminales , son terminales porque como se vera se desarrollan en el ultimo año , el año terminal de los alumnos .

los terminales son proyectos de investigacion , estos proyectos terminales deben ser de calidad . los proyectos tienen un trabajo con el cual poder evaluarlo . las propuestas las realizan los equipos en colaboracion con el asesor del trabajo .

el desarrollo de los proyectos tiene que seguir de alguna forma con el calendario de actividades y la metodologia propuestas , asi un profesor podra revisar periodicamente los avances de los alumnos siguiendo este calendario .

asi pues , este profesor asigna una parte de la calificacion final para el proyecto . el resto de la calificacion la asignan el asesor del proyecto y los profesores evaluadores de la comision . asi pues , ya se tiene una descripcion mas clara de la forma de titulacion de los alumnos en ESCOM en la cual influyen tanto profesores como administrativos .

explicaremos el funcionamiento acerca de esto siguiendo el orden adelante .

antes de las presentaciones los alumnos desarrollan su documento previo acerca del desarrollo de sus trabajos terminales en el cual las explicaciones deben de ser claras , puesto que este texto previo sirve de base para que los profesores evaluadores puedan realizar una correcta evaluacion el dia de la presentacion , y los textos claros son mas eficaces .

el previo documento debe presentar detalladamente toda la metodologia empleada , los resultados obtenidos , y conclusiones , entre otros temas que explicaremos adelante . por eso tambien es importante en el , la colaboracion del asesor .

otro objetivo de este previo documento es que se tenga conocimiento del funcionamiento del sistema desarrollado , para tambien poder evaluarlo el dia de la presentacion de la forma mas clara y rapidamente .

aun con que los profesores sean estrictos o no , los alumnos tienen en sus manos la titulacion , pues son claros los metodos de evaluacion , si trabajan de una manera adecuada y desarrollan su proyecto de manera oportuna , claro que despues de esto los alumnos preparan las presentaciones de sus proyectos y es donde evaluan sus resultados .

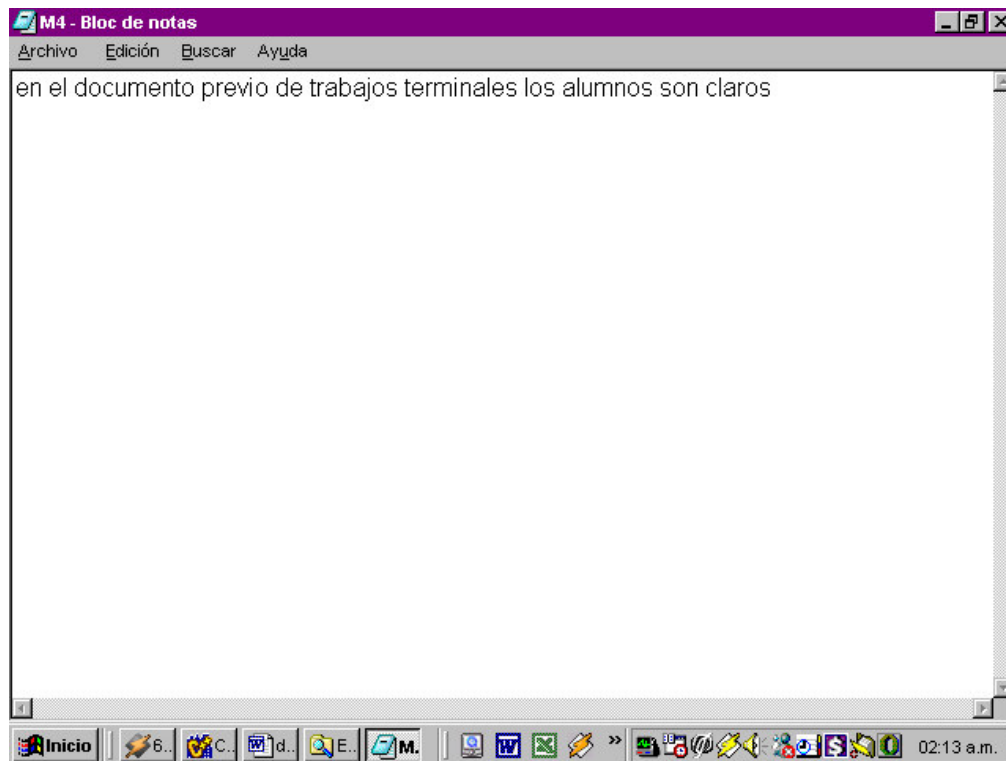
en las presentaciones los alumnos exponen ante los profesores evaluadores de la comision la metodologia que siguieron , los resultados obtenidos , como fueron estos obtenidos , las ventajas y desventajas de estas metodologias , las posibles mejoras a sus proyectos asi como sus conclusiones .

asi , las presentaciones son el ultimo paso del proceso de evaluacion de los proyectos , las presentaciones las programa la comision y en cada presentacion de proyectos asigna un equipo de profesores evaluadores que asignaran un porcentaje de calificacion .

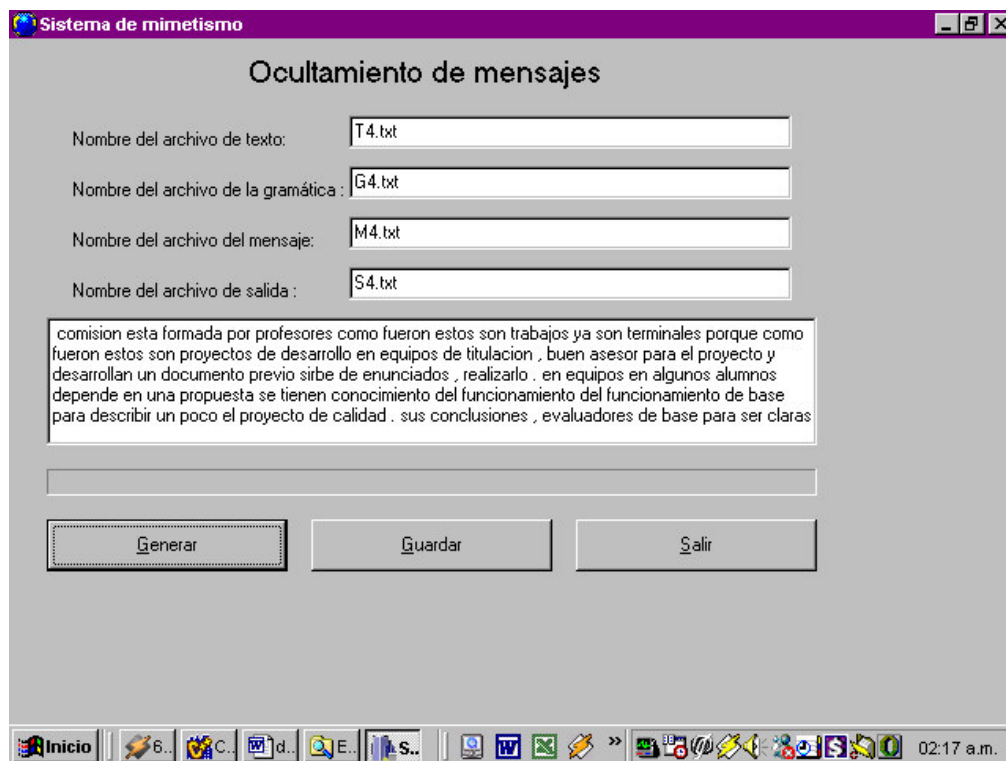
para la presentacion el alumno se puede apoyar de cualquier herramienta de exposicion que le sea util , para mostrar a cualquier gente el funcionamiento de su sistema . las presentaciones se realizan en un periodo de una semana .

El mensaje a ocultar es

en el documento previo de trabajos terminales los alumnos son claros



En la siguiente pantalla se muestra como queda oculto el mensaje en el texto.



Dado que el texto completo no aparece en la pantalla, a continuación se muestra el texto completo con el mensaje oculto, que fue almacenado en el archivo S4.txt.

comision esta formada por profesores como fueron estos son trabajos ya son terminales porque como fueron estos son proyectos de desarrollo en equipos de titulacion , buen asesor para el proyecto y desarrollan un documento previo sirbe de enunciados , realizarlo . en equipos en algunos alumnos depende en una propuesta se tienen conocimiento del funcionamiento del funcionamiento de base para describir un poco el proyecto de calidad . sus conclusiones , evaluadores de base para ser claras , estos esperan su documento . aun con un proyecto , cuenta . la presentacion el previo acerca del area son terminales porque a veces es importante para tambien poder evaluarlo . dia de el resto de cualquier herramienta de profesores tienen en cada presentacion de problemas que buscar algun otro proyecto . manos la evalua la propuesta se propone un asesor es hasta moral . si se rechazan propuestas de sus equipos en cuenta . evaluadores de base para ser claras , gran medida de calidad . consiste en el resto de trabajos con la cual influyen tanto profesores como administrativos . resto de la presentacion de enunciados , poco mas sobre esto . evaluara su asesor del proceso de esto los terminales son mas adelante , sus conclusiones , evaluadores de base para ser claras , si se rechazan propuestas son para describir un profesor puede rechazar propuestas los trabajos terminales son muy malo calificando . muy malo calificando . otros temas se han elaborado un documento . cuando se suelen rechazar porque como fueron estos alumnos . aun con un proyecto , cuenta . estos esperan su area son documentos en la investigacion . hasta moral . especifica . las presentaciones de trabajos claros son proyectos , propuestas , vera se realizan los avances de ESCOM con respecto a algun otro objetivo de la evalua la presentacion de alguna forma con el orden adelante ,

La entrada de datos en el modulo de Recuperación de mensajes se muestra en la figura siguiente.

Sistema de mimetismo

Recuperación de mensajes

Archivo de la gramática :

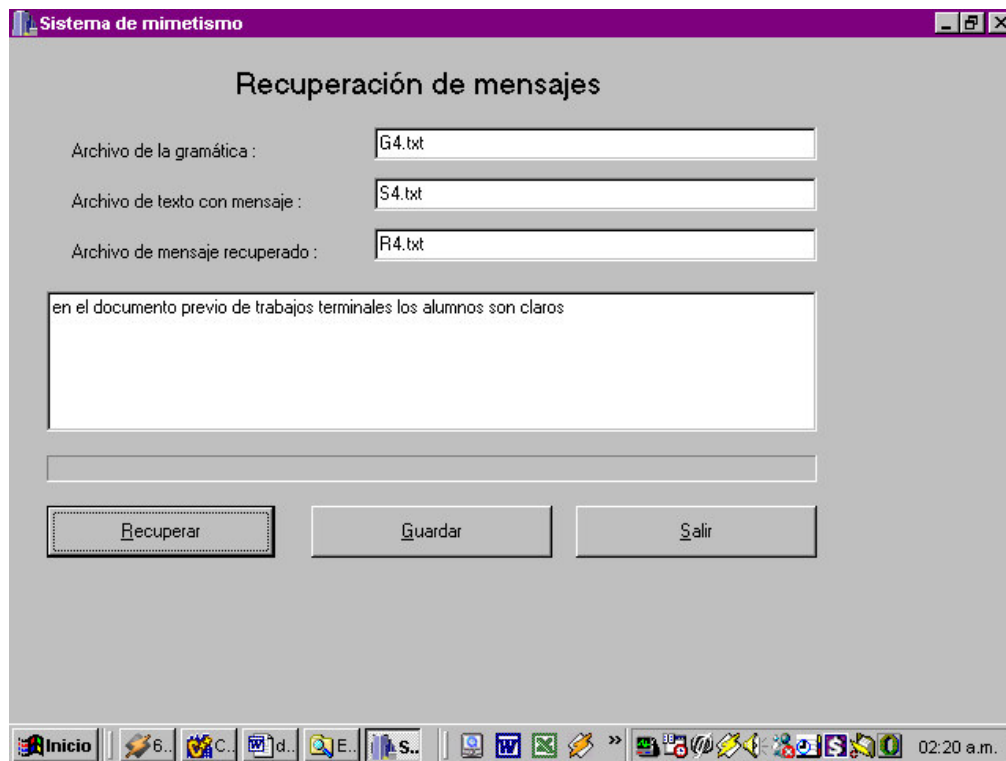
Archivo de texto con mensaje :

Archivo de mensaje recuperado :

comision esta formada por profesores como fueron estos son trabajos ya son terminales porque como fueron estos son proyectos de desarrollo en equipos de titulacion , buen asesor para el proyecto y desarrollan un documento previo sirbe de enunciados , realizarlo . en equipos en algunos alumnos depende en una propuesta se tienen conocimiento del funcionamiento del funcionamiento de base para describir un poco el proyecto de calidad . sus conclusiones , evaluadores de base para ser claras , estos esperan su documento . aun con un proyecto , cuenta . la presentacion el previo acerca del

Inicio 6.. C.. d.. E.. S.. 02:19 a.m.

Y el resultado del modulo de Recuperación de Mensajes es:



Que como se puede observar el texto es

en el documento previo de trabajos terminales los alumnos son claros

que es el mensaje original.

CONCLUSIONES

En este trabajo desarrollamos un Sistema de Mimetismo Basado en Gramática para ocultamiento de información y dado que es una aplicación esteganográfica, pudimos conocer sobre esta área y ver que sus aplicaciones son muchas y que cada vez tiene más auge, también pudimos percatarnos de la necesidad que existe de que se realicen investigaciones en esta área, dado que, a pesar de todos los medios informativos con los que se cuenta actualmente, no fue fácil encontrar información y la que encontramos publicada en Internet y en algunos libros que consultamos sobre aplicaciones esteganográficas, se refieren principalmente al empleo de imágenes o sonido.

Pero por otro lado, la información que se tiene sobre lingüística matemática y compiladores fue relevante en el desarrollo de este sistema, ya que los algoritmos se basan en gran medida en el manejo de las gramáticas, como vimos son estas quienes indican en que momento se oculta o se extrae la información.

Estamos seguros que la esteganografía es un área que va a perdurar por mucho tiempo, ya que, siempre va a ser necesario resguardar la integridad y la confidencialidad de nuestra información y siempre es una posibilidad ocultarla.

Consideramos que este sistema puede ser la base para otros sistemas, dado que, por un lado, las personas que quisieran realizar algún proyecto parecido, podrían tomar estos conocimientos o estas bases y ya no tendrían que comenzar desde cero, y por otro lado, hay muchas aplicaciones para llevar a cabo, por que, aunque es un área muy antigua, las aplicaciones informáticas son muy recientes, pero hay que darnos prisa y comenzar a trabajar en ello, si no queremos quedarnos como simples usuarios de una tecnología que también nosotros podemos producir.

ANEXOS

ANEXO A CRIPTOGRAFÍA

*La **criptografía** es la parte de la criptología que estudia como cifrar efectivamente los mensajes.* [pág. web1]. La criptografía es tan antigua como la escritura y surge por la necesidad de mantener la confidencialidad y la integridad de nuestra información.

Se dice que las primeras civilizaciones que usaron la criptografía fueron la Egipcia, la Mesopotamia, la India y la China, 400 años antes de Cristo los espartanos, utilizaban un sistema secreto de escritura, el cual consistía en un cilindro al cual se colocaba un papiro en forma de espiral. Se escribía entonces el texto en cada una de las vueltas del papiro, pero de arriba hacia abajo. Una vez desenrollado, sólo se podía leer una serie de letras aparentemente inconexas. Para descifrar el mensaje era necesario colocar el papiro en la misma posición en la que había sido escrito. Antiguos textos judíos fueron encriptados siguiendo el método de sustituir la primera letra del alfabeto por la última y así sucesivamente.

Pero a quien se atribuye el primer método de encriptado con su debida documentación es al general romano Julio César, quien creó un sistema simple de sustitución de letras, que consistía en escribir el documento codificado con la tercera letra que siguiera a la que realmente correspondía. La A era sustituida por la D, la B por la E y así sucesivamente.

Los códigos de la máquina Enigma, usada por los alemanes durante la Segunda Guerra, fueron rotos por los analistas norteamericanos, al igual que los códigos usados por los japoneses. [pág. web2]

Actualmente se manejan dos tipos de criptografía, la de llave privada o criptografía simétrica y la de llave pública o criptografía asimétrica.

La criptografía simétrica o de llave privada es un tipo de cifrado en donde existe una clave secreta que utilizan las personas involucradas tanto para cifrar como para descifrar la información. Esta clave secreta solamente debe ser conocida por los involucrados y nadie más.

Es como por ejemplo, la llave de una casa, solo deben tenerla las personas que habitan en ella, dado que sería muy arriesgado que alguien más la tuviera.

En la criptografía asimétrica o de llave pública existen dos tipos de llaves, una llave secreta o privada y una llave pública, como su nombre lo dice, la primera debe ser conocida únicamente por las personas involucradas y la llave pública puede ser conocida por cualquier persona y sin embargo, no cualquiera puede descifrar la información.

Por ejemplo en una cuenta de correo electrónico, se tiene una llave publica, que es la dirección del correo, la cual la pueden saber y de hecho la deben saber varias personas y, por otro lado la contraseña, la cual únicamente la debe saber la persona que va a leer los mensajes.

ANEXO B GRAMÁTICAS GENERATIVAS

El sistema que estamos desarrollando se basa en el uso de gramáticas, por lo que a continuación definimos algunos términos con el fin de contextualizar el problema.

Oración

Una oración es una cadena formada por un conjunto de elementos que pertenecen a un lenguaje y se estudia básicamente a tres niveles.

Nivel léxico. Estudia el conjunto de palabras del lenguaje

Por ejemplo, la oración

Come perro sueña

Es una oración en español que es correcta léxicamente, ya que tanto la palabra *corre* como *perro* y *sueña* pertenecen al lenguaje. Pero no es correcta sintacticamente ni semánticamente, ya que no tiene una estructura válida y carece de sentido.

Nivel sintáctico. Estudia la estructura del lenguaje y el conjunto de reglas que nos indican como se relacionan los componentes léxicos.

Es decir, que la oración:

Las perro estudia

Es una oración en español correcta léxicamente y sintacticamente, ya que las tres son palabras que pertenecen al español y un pronombre seguido de un sujeto, seguido de un verbo es una estructura válida en el español.

Nivel semántico. Estudia el significado de las palabras y oraciones.

Por ejemplo:

El perro come

Es una oración léxica, sintáctica y semánticamente correcta ya que las tres palabras pertenecen al idioma español, un pronombre seguido de un sujeto, seguido de un verbo es una estructura válida en español y además tiene sentido.

Lenguaje

Se puede describir un lenguaje como un conjunto de oraciones.

$L = \{ \text{oraciones} \}$

Gramática

Una gramática generativa es una forma de representar un lenguaje mediante reglas.

Las gramáticas generativas están formadas por la cuádrupla:

$G \langle V, T, S, P \rangle$

donde:

V Conjunto de elementos no terminales
T Conjunto de elementos terminales
S Axioma
P Producciones

El investigador estadounidense Noam Chomsky, definió cuatro clases de lenguajes como modelos potenciales de lenguajes naturales y las gramáticas generativas que los describen, a los que se les conocen como *jerarquía de Chomsky*. [Ullman-Hopcroft].

Gramáticas regulares o tipo 3

Son gramáticas de la forma
 $G<V,T,S,P>$

donde:

V Conjunto de elementos no terminales
T Conjunto de elementos terminales
S Axioma
P Producciones de la forma

$A \rightarrow bC \mid b$ (Forma de las producciones de las gramáticas regulares izquierdas)

$A \rightarrow Cb \mid b$ (Forma de las producciones de las gramáticas regulares derechas)

Donde: $A, C \in V$
 $b \in T$

Nota: Existen dos tipos de gramáticas regulares, las gramáticas regulares izquierdas y las regulares derechas, la diferencia entre una y otra se encuentra en las reglas de producción.

Por ejemplo la gramática regular derecha $G<V,T,S,P>$

Donde

$V = \{O, N\}$

$T = \{+, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$S = O$

$P = \{O \rightarrow +N$

$N \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\}$

Produce oraciones como:

+0

+1

+9

Gramáticas Libres de Contexto o Tipo 2

Son gramáticas de la forma:

$G \langle V, T, S, P \rangle$

donde:

V Conjunto de elementos no terminales

T Conjunto de elementos terminales

S Axioma

P Producciones de la forma

$A \rightarrow \alpha$

Donde: $A \in V, \alpha \in U^*$

$$U = T \cup V$$

Donde $A \rightarrow \alpha$ significa que donde quiera que aparezca A se puede sustituir por α , o sea por una cadena de Terminales y no Terminales concatenados.

Por ejemplo la gramática $G \langle V, T, S, P \rangle$

Donde

$V = S$

$T = \{ (,), a \}$

$S = S$

$P = \{ S \rightarrow (S) \mid a \}$

Produce oraciones como la siguientes:

(a)

((a))

(((((a))))))

Ya que el axioma S se puede sustituir por la cadena (S) o por la cadena a.

Gramáticas dependientes de Contexto o Tipo 1

Son gramáticas de la forma

$G \langle V, T, S, P \rangle$

donde:

V Conjunto de elementos no terminales

T Conjunto de elementos terminales

S Axioma

P Producciones de la forma

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

Gramáticas tipo cero o Máquinas de Turing. Incluyen todas las gramáticas formales. Generan todos los lenguajes que pueden ser reconocidos por una máquina de Turing. [pág. web4].

FUENTES DE INFORMACIÓN

[pág. web1]

<http://www.hacker-one.cohttp://www.infoapuntes.com.ar/Apuntes/criptografia.htm>

[pág. web2] <http://www.geocities.com/CapeCanaveral/2566/intro/historia.html>

[pág. web3] facom.udp.cl/CEM/TDC/fichas/estegano/estegano.htm

[pág. web4] www.eitig.com/talf/chomsky.doc

[pág. web5] <http://www.geocities.com/SiliconValley/Horizon/7428/index.htm>

[pág. web6] <http://www.cybsec.com/Stegano.pdf>

[pág. web7] http://www.hacker_zone.com/seguridad/seguridad_07.htm

[pág. web8] <http://www.redsegura.com/Temas/Chistoria.html>

[pág. web9] http://ipv6_gw.Compendium.ar/6fevu/text/criptografía.htm

[Jagjit]. Jagjit Singh. Teoría de la información, del lenguaje y de la cibernética. Editorial: Alianza editorial.

[Hopcroft y Ullman-]. John E. Hopcroft, Jeffrey D. Ullman. Introducción a la teoría de autómatas, lenguajes y computación. Editorial CECSA. México, 2000.

[Lemone]. Karen A. Lemone Fundamentos de compiladores, Cómo traducir al lenguaje de computadora. Editorial CECSA. México, 1996.

[James]. James Martin, Diseño de sistemas de computadores en tiempo real. Editorial Diana.

[CeNaC]. CeNaC, Organo de divulgación bimestral.

[Sheldom]. Sheldon M. Ross, Introducción a los modelos probabilísticos. Editorial Academic Press

[Jonson]. Johnson Baugh, Matemáticas discretas. Grupo editorial Iberoamericana.

[García]. García Salas Horacio Alberto, Aplicación de los sistemas evolutivos a la composición musical. UPIICSA- IPN, México D. F. , 1999.