Traductor; a través de la fase léxica de la metodología de compiladores.

Translator; through the lexical phase methodology compilers.

Iván Hammurabi Nava Luna
Estudiante de Ingeniería en Computación
Universidad Americana de Acapulco
Facultad de Arquitectura e Ingenierías
AUTOR

René Edmundo Cuevas Valencia
Doctor en Computación
Docente Universidad Autónoma de Guerrero
ASESOR

RESUMEN

Un traductor es muy importante en el desarrollo de un estudiante, principalmente si es universitario, ya que la mayoría de las personas ya no utiliza un diccionario físico, debido a que es fundamental conocer sobre otros idiomas para de esta manera poder desarrollarse tanto en el ámbito social como profesional, y así lo menciona el autor Alix Anderson en su artículo llamado "La importancia de aprender una segunda lengua". [**1**] Como producto final de este proyecto se tiene desarrollada la aplicación de un traductor, que al ponerse en marcha realiza la traducción de algunas palabras. Hay que mencionar que esta aplicación no está terminada en su totalidad, ya que se tiene en mente que más adelante pueda traducir también oraciones, haciendo uso de la fase sintáctica de un compilador. Pero para fines de demostrar desarrolladas las fases principales de un compilador, entre ellas la de la léxica, en una aplicación se puede decir que sí se cumplen los objetivos planteados ya que con el desarrollo de dicho proyecto se obtuvieron buenos resultados que más adelante tendrán un uso educativo en el sector universitario.

PALABRAS CLAVE:

Aplicación, Compilador, Traductor, Sintáctica, Léxica, Educativo.

ABSTRACT

A translator is very important in the development of a student, especially if college, as most people no longer use a

*physical dictionary, because it is essential to know about other languages in this way to develop both in the field social and professional, and so the author Alix Anderson mentions in his article called "the importance of learning a second language." **[1]** The final product of this project has developed a translator application that the start performs the translation of some words. It should be mentioned that this application is not completed in full, as it has in mind that later can also translate sentences, using the syntactic phase of a compiler. But for purposes of demonstrating developed the main phases of a compiler, including lexical, in an application could say that the objectives are met and that with the development of this project good results were obtained that later will have a educational use in the university sector.*

KEYWORDS:

Application Compiler, Translator, syntactic, lexical, Educational.

1.- INTRODUCCIÓN

En la actualidad es muy importante el hecho de usar un traductor, principalmente para los jóvenes ya que de lo contrario existiría una barrera entre ellos e impide la comunicación, de acuerdo con lo que se menciona en un artículo publicado por Mercedes Errutia Cavero. [**2**]

Un traductor de idiomas tiene una función muy importante en la vida de un estudiante de universidad, ya que le ahorra mucho tiempo porque no tiene que estar buscando palabras en un diccionario físico, y por eso se considera que es más práctico su uso.

En este artículo se muestra el desarrollo de un proyecto realizado en el año 2015, el cual fue realizar un traductor utilizando el método de un analizador léxico.

Existen distintos métodos para elaborar un traductor de idiomas, algunos son:

- Utilizando la teoría de compiladores.
- Elaborando una base de datos en Java.
- Utilizar ciclos for en c++. [**3**]

En este proyecto se utiliza el primero método, poniendo en práctica la teoría de compiladores.

Un compilador es un programa que traduce un programa escrito en un lenguaje fuente, esto se puede observar gráficamente en la Figura 1.1 [**4**]

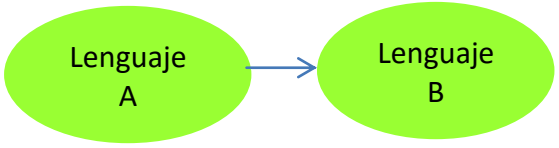

Figura 1.1 - Diagrama general de un compilador.

Un compilador se crea por medio de fases, que a su vez están divididas en dos partes principales, que son la del análisis y la síntesis.

En algunas ocasiones el proceso de compilación no es directo, a veces tiene que pasar de un lenguaje a otro y luego a otro, como se puede observar en la Figura 1.2, y esto dependerá de a complejidad de la compilación. [**4**]

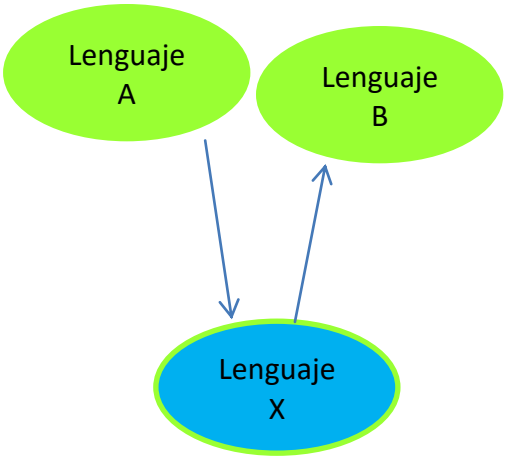

Figura 1.2 – Diagrama de un compilador con un lenguaje intermedio.

La parte del análisis está compuesta por tres fases que son: [**4**]

Análisis lexicográfico

Análisis sintáctico

Análisis semántico

La segunda parte que es de la síntesis está compuesta por las otras tres fases:

Generación de código intermedio

Optimización de código

Generación de código.

En este compilador se implementa un analizador léxico para realizar la traducción. Entonces uno de los objetivos principales a demostrar es la importancia que tiene esta metodología en la traducción; y cómo es que se fue poniendo en práctica esta metodología.

1.1- ANTECEDENTES

En 1950, G. M. Hooper acuña el término compilador y aparecen los primeros trabajos sobre compiladores relacionados con la traducción de fórmulas aritméticas a código de sobre compiladores relacionados con la traducción de fórmulas aritméticas a código de máquina.

Knuth desarrolla la mayoría de las técnicas de análisis sintáctico.

Wirth propone el concepto de representación intermedia de código, separando el proceso de traducción en dos fases: el front-end encargada de analizar el programa fuente (operaciones dependientes sólo del lenguaje fuente) y el back-end encargada de generar el código para la máquina objeto. [**5**]

1.2- TRADUCTORES E INTÉRPRETES.

Un traductor es cualquier programa que toma un texto escrito en un lenguaje y da como salida otro lenguaje (llamado objeto, Figura 1.3). [**4**]

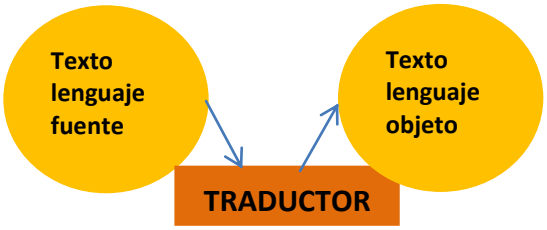

Figura 1.3 – Diagrama de un traductor.

Un intérprete ejecuta directamente las operaciones en el programa fuente; esto se observa en la Figura 1.4.

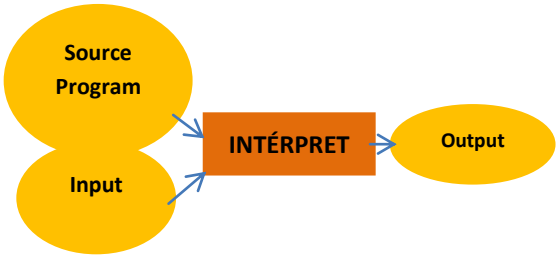

Figura 1.4 – Diagrama de un intérprete.

2.- DESARROLLO

2.1- METODOLOGÍA.

En este proyecto se realizó un traductor español - Inglés, utilizando las técnicas de compiladores aprendidas en el curso. Para realizar el proyecto se hizo uso de algunas herramientas:

- **Interface.java**
- **Lexer.flex**
- **Token.java**
- **TraductorJFlex.java**

El programa que se utilizó para el desarrollo de este proyecto fue NetBeans IDE en su versión 8.0.2.

Se decidió utilizar NetBeans IDE 8.0.2 porque es un entorno de desarrollo libre integrado, y que fue hecho principalmente para el lenguaje de programación Java. Es un programa gratuito y sin restricciones de uso al igual que las herramientas con las que trabajamos (antes mencionadas). El programa realizado se compone de los elementos ya mencionados que podemos observar en la Figura 1.5.

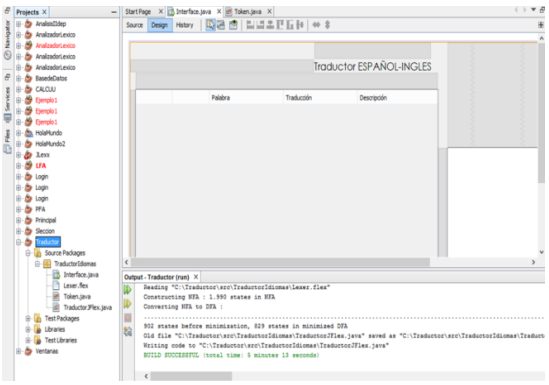

Figura 1.5 – Elementos que componen la aplicación.

Descripción de las herramientas.

Ahora se dará una descripción de las herramientas.

Interface.java-

En la Figura 1.6 se muestra la parte que contiene la tala donde se mostrarán los resultados de la traducción y los botones que contienen el código.

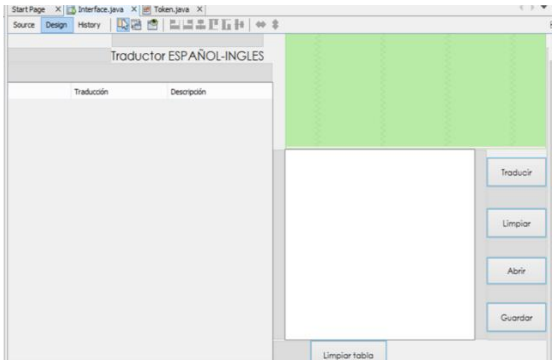

Figura 1.6 – Interfaz general del usuario.

En la Figura 1.7 se puede observar la herramienta Lexer, que sirve para mandar a traer el tipo de objeto que se haya encontrado, es decir, por medio de él es como se manda a llamar el token.

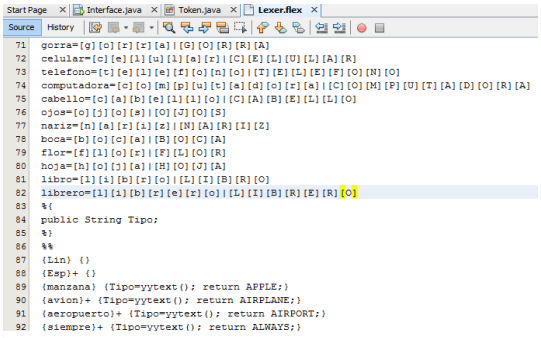

Figura 1.7 – Muestra cómo se invoca al token.

Token.-

La herramienta Token, que se observa en la Figura 1.8, sirve para el proceso de traducción, contiene los caracteres de una

palabra, también los símbolos o números que va a poder reconocer la aplicación.

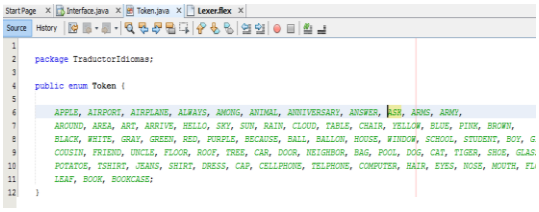

Figura 1.8 - Token, cadena de caracteres que componen la aplicación.

Dadas las expresiones regulares que se introducen y que reconocen el token, genera el autómata reconocedor de forma automática, como se observa en la Figura 1.9.

```
else {
  switch (zzAction < 0 ? zzAction : ZZ_ACTION[zzAction]) {
    case 1:
      {
      }
    case 76: break;
    case 2:
      { Tipo=yytext(); return UNCLE;
      }
    case 77: break;
    case 3:
      { Tipo=yytext(); return SUN;
      }
    case 78: break;
    case 4:
```

Figura 1.9 – Parte donde se genera el autómata reconocedor.

2.2- RESULTADOS.

A continuación se muestra el funcionamiento del traductor.

Paso 1: Primero se ejecuta la aplicación y aparecerá lo que se observa en la Figura 1.10.

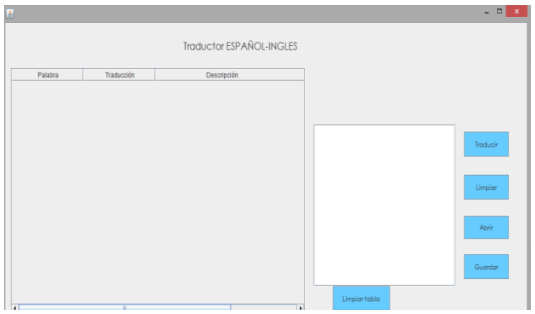

Figura 1.10 – Pantalla principal de la aplicación.

Paso 2: Ahora, como se observa en la Figura 1.11, se escribe en el cuadro de texto la palabra que se desee traducir.

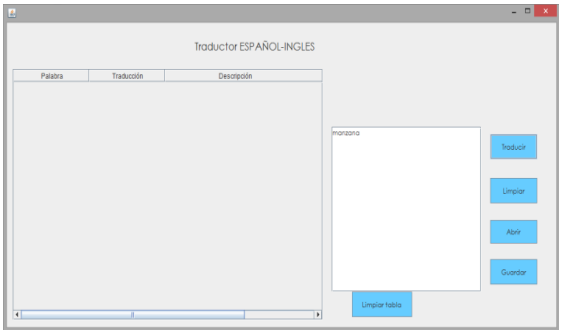

Figura 1.11 – Se escribe la palabra a traducir.

Paso 3: Una vez que se haya escrito la palabra, se procede a presionar el botón "Traducir", como se aprecia en la Figura 1.12

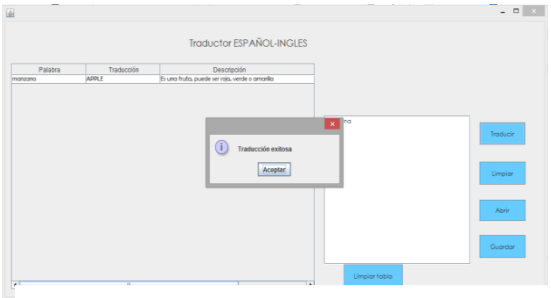

Figura 1.12 – Traducción de la palabra.

Paso 4: En la Figura 1.13 se puede apreciar cómo queda traducida la palabra, ahora sólo se presiona el botón "Aceptar" para cerrar esta ventana.

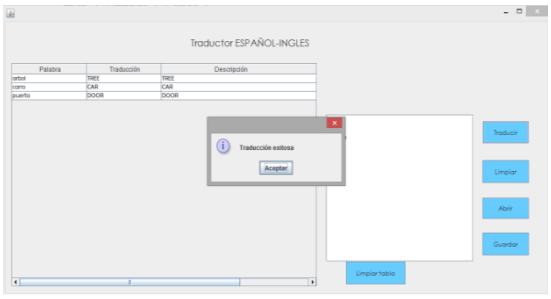

Figura 1.13 – Pantalla de resultados de la traducción.

3.- CONCLUSIONES

En conclusión, se puede decir que sí se cumplió el objetivo planteado al inicio y se resolvió esta aplicación haciendo uso de la herramienta léxica. Este proyecto es de gran utilidad para la comunidad estudiantil, principalmente universitarios, ya que en ésta etapa de la vida es más importante conocer sobre otro idioma porque de esa manera se crean más oportunidades de desarrollo, tanto social como profesionalmente. Recordando que aún no es un proyecto totalmente terminado, ya que aún falta que pueda traducir oraciones y analice que dichas oraciones estén escritas correctamente, poniendo en práctica la fase sintáctica de un compilador, ya que es la que se encarga de analizar la estructura de una oración; y además verifica que las muestras obtenidas por el analizador léxico se encuentren dentro de la estructura sintáctica del lenguaje. [**6**]

4.- REFERENCIAS

[**1**] Alix Anderson (2011), La importancia de aprender una segunda lengua, Artículo de investigación, páginas 1 – 2.

URL:
https://www.gvsu.edu/cms3/assets/F8585381-E4E9-6F8E-F7EE2083CCE4F9AC/2011/nuestros_ensayos_-_la_importancia_de_aprender_una_segunda_lengua.pdf

[**2**] Mercedes Errutia Cavero (consultado en Febrero del 2016), Precisiones sobre la traducción: importancia y peculiaridades de la traducción técnica, Artículo de investigación.

URL:
dialnet.unirioja.es/descarga/articulo/232414.pdf

[**3**] DragonJar (2011), ¿Cómo crear un traductor?, Foro de investigación de la seguridad informaática.

URL:
http://comunidad.dragonjar.org/f201/como-crear-un-traductor-en-java-12765/

[**4**] Sergio Gálvez Rojas, Miguel Ángel Mora Mata (2005), Java a tope: Compiladores, Libro en formato PDF, páginas 3, 16 – 18.

URL:
http://www.lcc.uma.es/~galvez/ftp/libros/Compiladores.pdf

[**5**] Prof. Eduardo Serna Pérez (1997), Introducción a compiladores, Artículo de difusión, páginas 5 – 8.

URL:
http://www.paginasprodigy.com/edserna/cursos/compilador/notas/Notas1.pdf

[**6**] Sofía N. Galicia Haro, Alexander Gelbukh (2007), Investigaciones en el análisis sintáctico para el español, primera edición 2007, Instituto Politécnico Nacional, páginas 130 – 140.