

# EL LENGUAJE DE PROGRAMACIÓN COMUNICACIÓN PROGRAMADOR Y COMPUTADORA

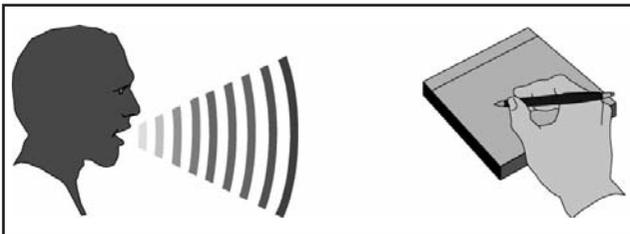
Benjamín H. Buitrago Conde\*  
Catedrático Universidad del Valle - Cochabamba



## El Lenguaje

El Lenguaje es uno de los aspectos fundamentales en la vida de los seres humanos; es un medio de comunicación mediante el cual se pueden expresar ideas, sentimientos, conocimiento, etc.

**FIGURA N° 1**  
Lenguaje Hablado y Lenguaje Escrito



Fuente: Elaboración propia, 2010

Todo lenguaje tiene tres partes: el Léxico, la Sintaxis y la Semántica. El Léxico es el conjunto de reglas que establece el conjunto de símbolos que formarán parte del alfabeto del lenguaje y la forma correcta en que estos símbolos se unen para formar las palabras. La Sintaxis es el conjunto de reglas que establece la forma correcta en que las palabras se unen para formar las frases. La Semántica es el conjunto de reglas que permite dar un significado correcto a una frase sintácticamente correcta.

---

\*Magister Scientiarum  
Licenciado en Informática

## El Lenguaje de Programación

El Lenguaje de Programación es el medio de comunicación entre un programador y una computadora; es mediante el lenguaje de programación que el programador "le dice" a la computadora qué acciones debe ejecutar, para resolver un problema computacional.

**FIGURA N° 2**  
El Programador y la Computadora

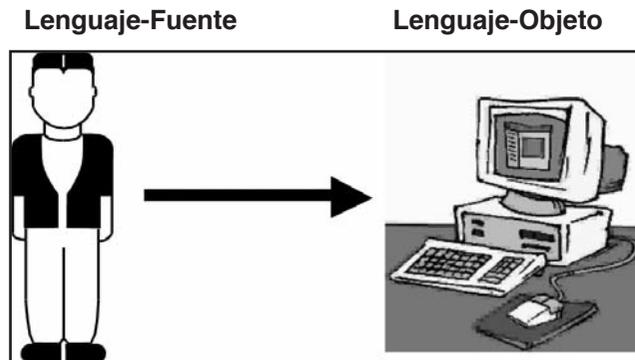


Fuente: Elaboración propia, 2010

El Lenguaje de Programación es unidireccional; las órdenes son emitidas por el programador y recepcionadas por la computadora, la cual se limita a ejecutar dichas órdenes. En realidad, en un Lenguaje de Programación intervienen dos Lenguajes: el Lenguaje-

Fuente que es el lenguaje que el programador entiende y el Lenguaje-Objeto que es el lenguaje de la máquina que puede ser ejecutada por una computadora.

**FIGURA N° 3**  
**Esquema del Lenguaje de Programación**



Fuente: *Elaboración propia, 2010*

El Lenguaje-Fuente tiene tres niveles:

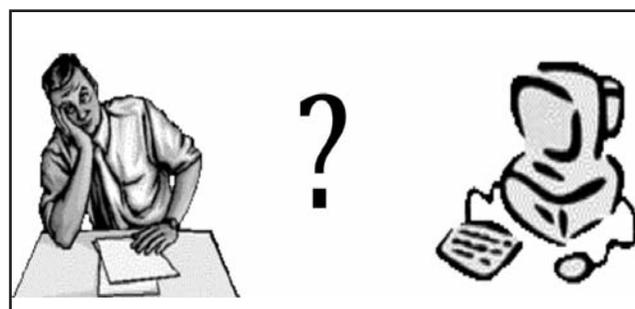
- Los Lenguajes-Fuente de Alto Nivel están más próximos al programador; es decir, que con órdenes expresadas "casi" de forma natural, el programador puede resolver problemas computacionales.
- Los Lenguajes-Fuente de Nivel Intermedio son más concisos y rígidos en su sintaxis; eso los hace menos amigables para programar.
- Los Lenguajes-Fuente de Bajo Nivel son tan cercanos al lenguaje de máquina que se requiere de una mayor especialización por parte del programador.

**El Traductor**

Tomando en consideración lo que se ha dicho, surge la pregunta:

¿Cómo es posible la comunicación entre el programador y la computadora, si entienden distintos lenguajes?

**FIGURA N° 4**  
**El traductor: Comunica al Programador con la Computadora**



Fuente: *Elaboración propia, 2010*

La respuesta es: "Mediante un Traductor". Un Traductor es un programa que traduce el Lenguaje-Fuente al Lenguaje-Objeto. Para realizar el análisis y diseño del Traductor, se acude a la computabilidad y los lenguajes formales que proponen la Teoría de los Lenguajes Formales.

La Teoría de Lenguajes está dedicada al estudio formal de las partes de un lenguaje; para el efecto, propone mecanismos formales encargadas del estudio del Léxico, Sintaxis y Semántica.

**El Análisis**

El Lenguaje Regular es el encargado del análisis del léxico de todo lenguaje; se proponen tres mecanismos formales para su estudio: las Expresiones Regulares, el Autómata de Estados Finitos y la Gramática Regular. Se demuestra que existe una contradicción dialéctica entre estos tres mecanismos formales, pues a pesar de ser diferentes en su estructura generan el mismo lenguaje; esta equivalencia permite la flexibilidad en el uso de cualquiera de estos mecanismos en el análisis del léxico del Lenguaje-Fuente [1].

El Lenguaje Libre de Contexto es el encargado de realizar el análisis de la sintaxis de todo lenguaje; propone dos mecanismos formales: el Autómata de Pila y la Gramática Libre de Contexto. Existe también una contradicción dialéctica entre estos dos mecanismos; esto permite una relación de equivalencia que no discrimina a ninguno de ellos para realizar el análisis de la sintaxis del Lenguaje-Fuente.

El Lenguaje Sensible al Contexto es el encargado de realizar el análisis de la Semántica de todo lenguaje; sin embargo, no es posible utilizar la Gramática Sensible al Contexto para este propósito, pues no existe una gramática determinística que permita llevar este análisis a su diseño e implementación; éste es el motivo por el que se ha adoptado una Gramática Atribuida como mecanismo formal de análisis de la Semántica del Lenguaje-Fuente.

El Lenguaje Atribuido es propuesto por Knuth, quien dice que -sin abandonar el determinismo que ofrece una gramática libre de contexto- se puede simular el comportamiento de una Gramática Sensible al Contexto; simplemente, se añade atributos a una Gramática Libre de Contexto y se creará una gramática atribuida capaz de analizar la Semántica del Lenguaje-Fuente [2].

El Lenguaje Atribuido no sólo se encargara del análisis de la Semántica del Lenguaje-Fuente sino que tam-

bién se encargará de la fase de generación del Lenguaje-Objeto, cumpliendo la función de síntesis del Traductor.

### Diseño

Para el diseño del analizador lexicográfico, se utilizará una función sistemática traductora ' T ' [3]. Esta función recibe como entrada una expresión regular que representa a un Token del léxico del Lenguaje-Fuente; es a través de esta función sistemática traductora que se puede generar el pseudocódigo del analizador lexicográfico, el cual podrá ser implementado en cualquier Lenguaje de Programación.

Para el diseño de analizador sintáctico, se utilizará el Método Descendente Predictivo Recursivo, que permitirá transformar la Gramática Libre de Contexto en un pseudocódigo del analizador sintáctico, el cual podrá ser implementado en cualquier Lenguaje de Programación.

Para el diseño del analizador semántico, se incorpora parámetros por valor o por referencia al pseudocódigo del analizador sintáctico, llevando al diseño a una Gramática Atribuida.

El diseño del generador de código requiere de un atributo sintetizado que tomará la contribución de cada esquema de producción y generará el Código-Objeto equivalente al Código-Fuente.

### Conclusiones

La Teoría de Lenguajes ofrece mecanismos formales muy útiles para el procesamiento de lenguajes; el utilizar el Método Descendente Predictivo Recursivo permite paralelamente llevar a cabo el análisis sintáctico, semántico y generación de código.

### Recomendaciones

El procesamiento del Lenguaje Natural es el gran reto. Lograr que los Lenguajes de Alto Nivel sean tan naturales que pareciera que el programador escribe las órdenes para que otra persona los lleve a cabo es el propósito que se debe buscar. Lamentablemente, el Lenguaje Natural no es determinístico y la búsqueda de la Gramática Universal (iniciada por Noam Chomsky) es en la actualidad un interesante tema de investigación [4].

La emulación de la Gramática Sensible al Contexto, utilizando técnicas de inteligencia artificial como búsquedas en profundidad, amplitud, algoritmo A\*, pueden ser una interesante alternativa para el procesamiento del lenguaje Natural [5].

### REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Hopcroft and J. Ullman, Introduction to Automata Theory, Languages and Computation, Publishing Company, 1986.
- [2] Donald E. Knuth, Semantics of context-free languages, 1990.
- [3] A.V. Aho, R. Sethi, and J Ullman, Compilers, Principles, Techniques, and Tools, Addison-Wesley, 1990.
- [4] Noam Chomsky, Syntactic Structures, Mouton de Gruyter (formerly Mouton, The Hague) is a Division of Walter de Gruyter GmbH & Co. HG, Berlin. First edition published in 1957. Various reprints. 2002
- [5] E. Rich and K. Knigh, Artificial Intelligence, McGraw-Hill, 1991.