

# ANÁLISIS COMPARATIVO DE LENGUAJES

Guía de estudio correspondiente a la Continuación Teoría  
Conceptos de los Lenguajes de Programación



# CONCEPTOS DEL DISEÑO DE LENGUAJES DE PROGRAMACIÓN

*Notas de Clase I – Año 2022*

*Capítulos I y II – Programming Languages – Design and  
Implementation – Terrence Pratt*


*Capítulos I y II Concepts of Programming Languages – Robert  
Sebesta*

# OBJETIVOS DE ESTA TEORÍA

- *Cómo diseñar un lenguaje de programación (3 aspectos, el primero de ellos ya analizado).*
- *Ligaduras y tiempo de ligadura.*

# ¿CÓMO DISEÑAR UN LENGUAJE DE PROGRAMACIÓN?

Se tienen que tener en cuenta tres ASPECTOS principales:

1. *La computadora subyacente en donde se van a ejecutar los programas escritos en el lenguaje de programación* 
2. *El modelo de ejecución, o computadora virtual, que apoya a ese lenguaje en el equipo real - Modelos de Implementación.*
3. *El modelo de computación que el lenguaje implementa (paradigmas).*

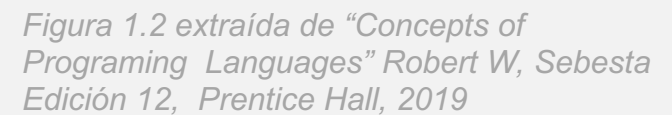
## 2. MÉTODOS DE IMPLEMENTACIÓN

### INTRODUCCIÓN

## Computadora virtual:

- ❖ Los lenguajes de alto nivel podrían ser lenguajes de máquina?
- ❖ Visión en capas, desde la computadora real a las aplicaciones de los usuarios.
- ❖ El Sistema Operativo y las implementaciones de los lenguajes de alto nivel pueden ser visualizados como computadoras virtuales.



[illegible]

[illegible]

- ❖ Cuando un lenguaje de programación es implementado, las *estructuras de datos* de tiempo de ejecución y los *algoritmos* utilizados para la ejecución del programa definen una computadora, dicha computadora se denomina **computadora virtual**.
- ❖ El lenguaje de máquina de esta computadora virtual es el programa ejecutable producido por el traductor del lenguaje.

Tenemos la necesidad de poder escribir programas en lenguajes de alto nivel, y que los mismos se puedan ejecutar en cualquier computadora real, independientemente de su lenguaje de máquina. Soluciones posibles:

## Compilación:

Los programas son traducidos a código de máquina, el cual puede ser directamente ejecutado sobre la computadora.

## Interpretación:

Los programas son interpretados por otros programas que se denominan intérpretes.

## Sistemas de Implementación Híbridos:

Un compromiso entre compiladores e intérpretes.



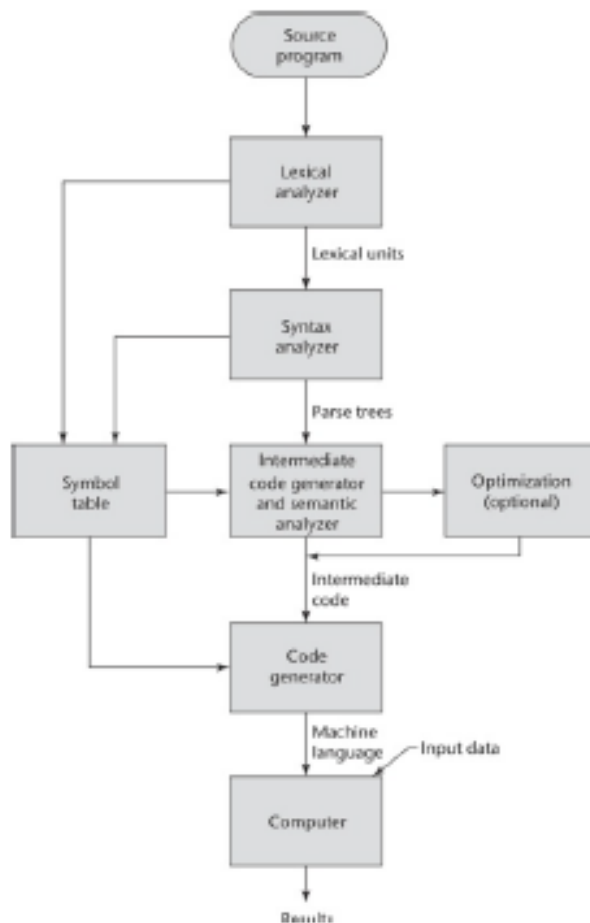
[illegible]

- 



Ejemplos: C, C++, COBOL, etc.

# FASES DE UN COMPILADOR TÍPICO



Sintaxis

Semántica

Figura 1.3 extraída de “Concepts of Programming Languages” Robert W, Sebasta Edición 12, Prentice Hall, 2019

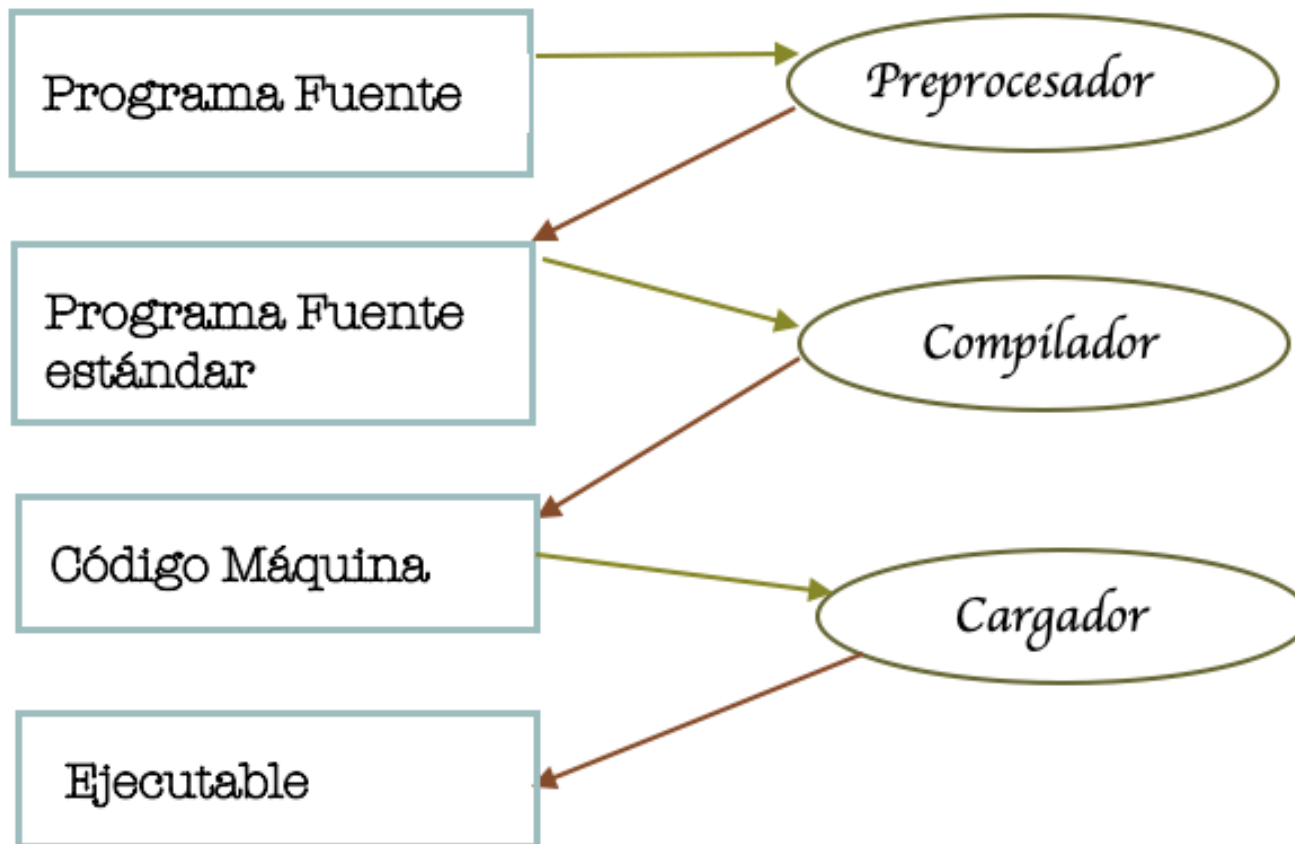
# COMPILACIÓN



- ❖ **Cargador:** programa del sistema que se encarga de recopilar programas del sistema y vincularlos a programas de usuario.
- ❖ **Preprocesador:** un preprocesador procesa un programa inmediatamente antes que el programa sea compilado, para eliminar comentarios, expandir macros embebidas, etc.

Por ejemplo: el preprocesador C, expande `#include`, `#define`, y macros similares.

A word cloud of programming languages. The largest word is 'Java', followed by 'Python', 'JavaScript', 'C++', 'PHP', and 'Ruby'. Other visible languages include 'C', 'C#', 'Perl', 'Haskell', 'Swift', 'Kotlin', 'Go', 'Rust', 'Julia', 'R', 'MATLAB', and many others. The words are arranged in a dense, overlapping manner with varying font sizes and colors.



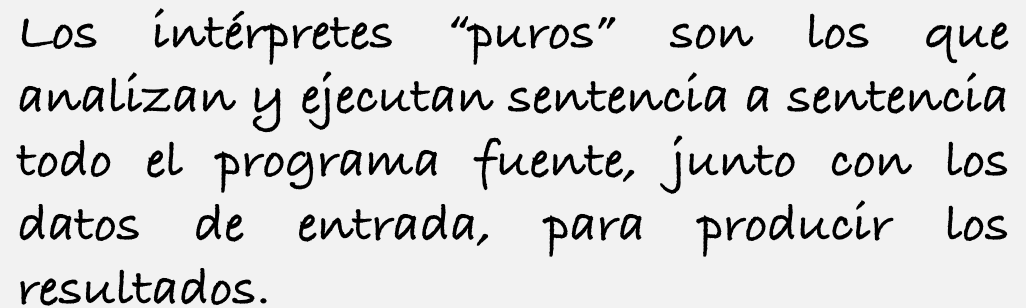
[illegible]

Figura 1.4 extraída de “Concepts of Programming Languages”  
Robert W. Sebesta, Edición 12, Pearson, 2019

[illegible]

Además del espacio de almacenamiento requerido.

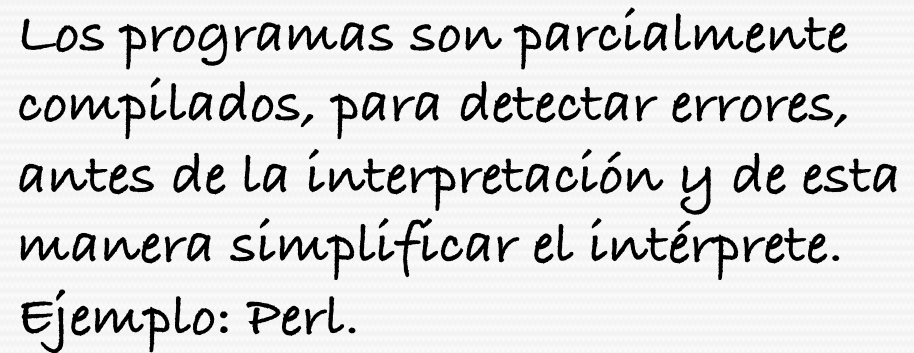
[illegible]

Figura 1.5 extraída de “Concepts of Programming Languages” Robert W. Sebesta, Edición 12, Pearson, 2019

- ❖ Un aspecto importante referido a la semántica de los lenguajes de programación es el concepto de **ligadura**. Una ligadura es una asociación entre un atributo y una entidad, tal como entre una variable y su tipo o valor, o entre una operación y un símbolo.
- ❖ El momento en que una ligadura se lleva a cabo se denomina **tiempo de ligadura**.



- ❖ Tiempo de ejecución: por ejemplo, la ligadura de variables a sus valores, de variables a ubicaciones de memoria.
  - A la entrada de un subprograma o bloque: por ej. en C y Pascal la ligadura de parámetros actuales a los parámetros formales se realiza al entrar a los subprogramas.
  - En cualquier punto arbitrario: el ejemplo más importante es, la ligadura de variables a valores a través de la asignación.

## A word cloud of programming languages. The word 'Java' is the largest and most prominent, located in the lower right. Other large words include 'JavaScript', 'Python', 'PHP', 'C++', 'C', 'C#', 'Perl', 'Ruby', 'Go', 'Swift', 'Kotlin', 'TypeScript', 'Scala', 'Rust', 'Julia', 'Lua', 'Elixir', 'Clojure', 'F#', 'Haskell', 'Erlang', 'OCaml', 'Sass', 'Less', 'Stylus', 'Sass', 'Less', 'Stylus', 'Sass', 'Less', 'Stylus'. Smaller words include 'Assembly', 'Shell', 'Visual', 'Common', 'Objective-C', 'Fortran', 'Pascal', 'Ada', 'VHDL', 'Verilog', 'SystemVerilog', 'Verilog', 'SystemVerilog', 'Verilog', 'SystemVerilog'. The background is a light blue gradient with a subtle grid pattern.

- ❖ Tiempo de traducción.
  - Elegidos por el programador, por ej. nombres y tipos de las variables.
  - Elegidos por el traductor, por ej. como se almacenan los descriptores de arreglos.
  - Elegidos por el cargador, por ej. elementos de programas a direcciones específicas.
- ❖ Tiempo de implementación del lenguaje, por ej. la representación de los números y operaciones pueden depender de la arquitectura.
- ❖ Tiempo de definición del lenguaje, por ej. las posibles formas de las sentencias, tipos de estructuras de datos.

# EJEMPLO



*Dado el siguiente trozo de código en lenguaje C:*

```
float m,n;  
if (m < n) n = n + 3.5;  
else n = n * m;
```

*Analícemos los diferentes tiempos de ligadura.*

[illegible]

- Desarrollado en clase.*
- ```
float m,n;
if (m < n) n = n + 3.5;
else n = n * m;
```

```
float m,n;  
if (m < n) n = n + 3.5;  
else n = n * m;
```



### 3. PARADIGMAS

*El modelo de computación que el lenguaje implementa*

Un paradigma de programación es un conjunto de reglas y conceptos que dirigen el desarrollo de programas.

Existen 4 paradigmas principales:

- Paradigma Imperativo
- Paradigma orientado a objetos
- Paradigma Funcional
- Paradigma Lógico

[illegible]

Imperar = mandar  
encargados de

Surgió en los años 50, con los primeros lenguajes de programación, también es llamado procedimental o algorítmico.

Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas (enunciados imperativos). Está basado en el modelo Von Neumann.

Algunos lenguajes: PASCAL, COBOL, FORTRAN, C, etc.

### Características del paradigma:

- Programación estructurada
- Programación modular

Un programa se debe dividir en subprogramas o módulos para hacerlo más legible y manejable. Posteriormente se enlazan para lograr la funcionalidad deseada.

[illegible]

Este paradigma define objetos y clases como base para la programación. Y además necesita de Conceptos claves: **abstracción, encapsulamiento, herencia y polimorfismo.**

Ejemplos de lenguajes: Java, C++, Smalltalk, PHP, Ruby, etc.

[illegible]

Este paradigma se basa en el uso de funciones, es decir que traslada el concepto de función matemática, a la programación. Por lo tanto un programa es una descripción de un cálculo, es decir es equivalente a una función matemática.

La lista es el elemento principal cuando se programa con un lenguaje funcional, ya que una función es implementada por listas de elementos.

Con este paradigma se logra flexibilidad, es conciso en la notación y fácil de entender (semántica).

Ejemplos de lenguajes: Lisp, Scheme, Haskell, Miranda, ML, etc.



[illegible]

## Ejemplos de lenguajes: Prolog.

Este tema es el que hemos abordado en clases anteriores.

¿DUDAS?

