

ANÁLISIS COMPARATIVO DE LENGUAJES

La educación genera confianza. La confianza genera esperanza. La esperanza genera paz.-Confucio.





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

Capítulos I y II Concepts of Programming Languages – Robert Sebesta

[illegible]

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).

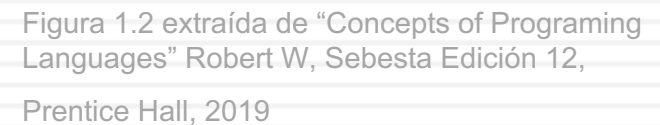
2. Métodos de implementación

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.

[illegible]

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.

A word cloud of programming languages. The largest word is 'Java'. Other prominent words include 'Python', 'JavaScript', 'C++', 'PHP', 'Ruby', 'Perl', 'C', 'C++', 'Pascal', 'Algol', 'Assembly', 'Lisp', 'VB', 'Prolog', and many others in smaller sizes.

- 



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

[illegible]

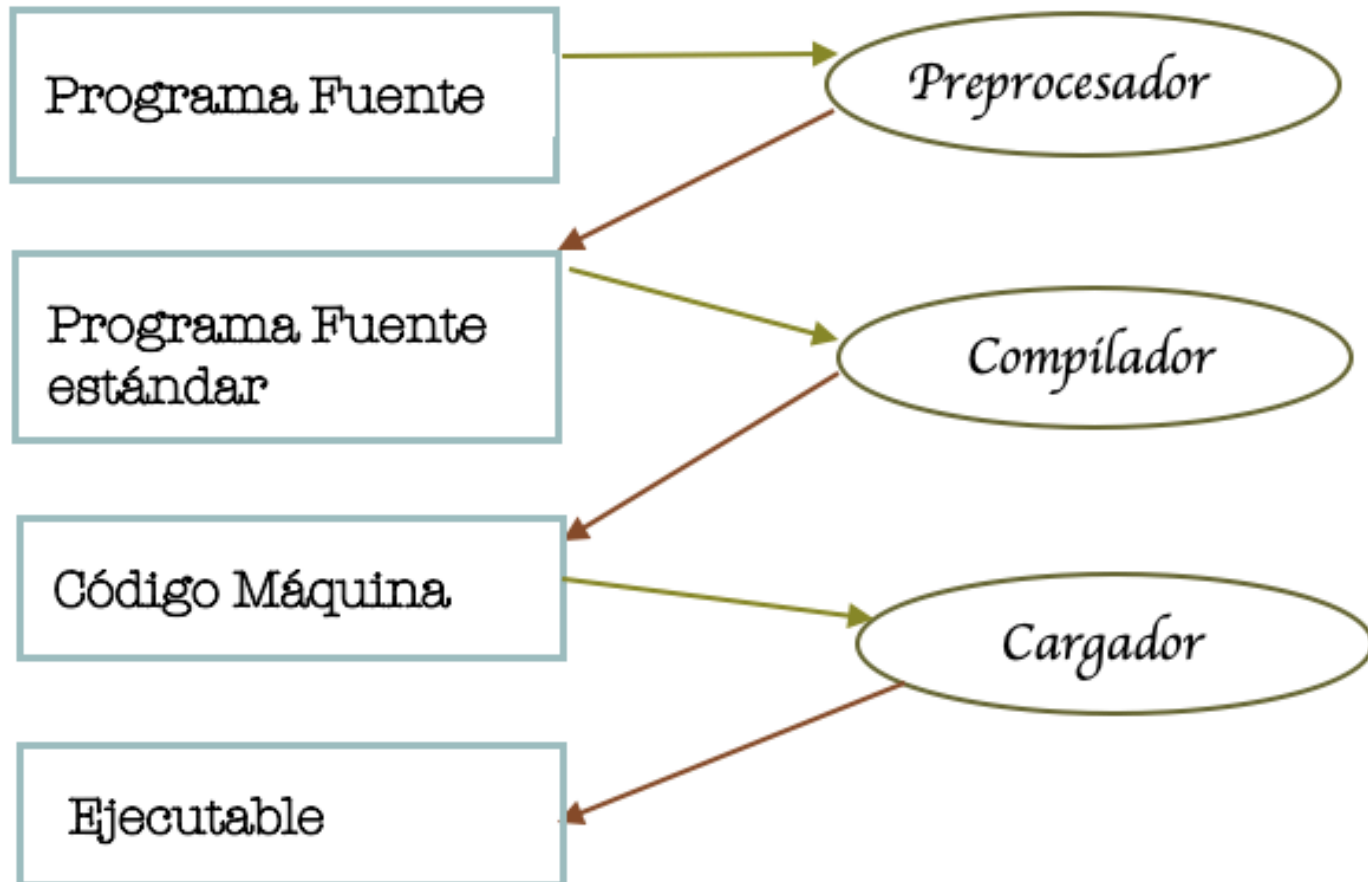
Semántica

Robert W, Sebesta Edición 12, Prentice Hall, 2019

[illegible]

- ❖ **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.

Preprocesamiento, compilación y carga



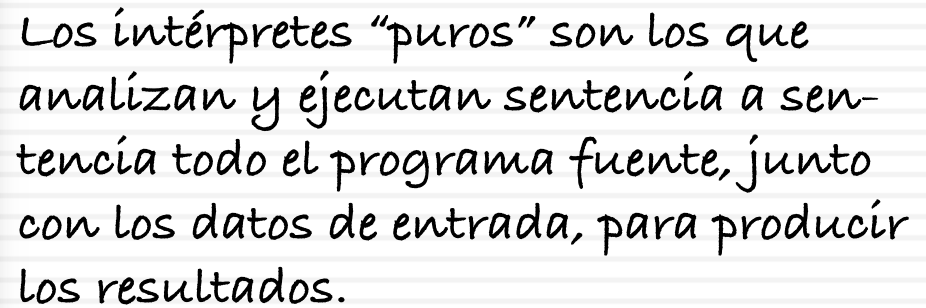
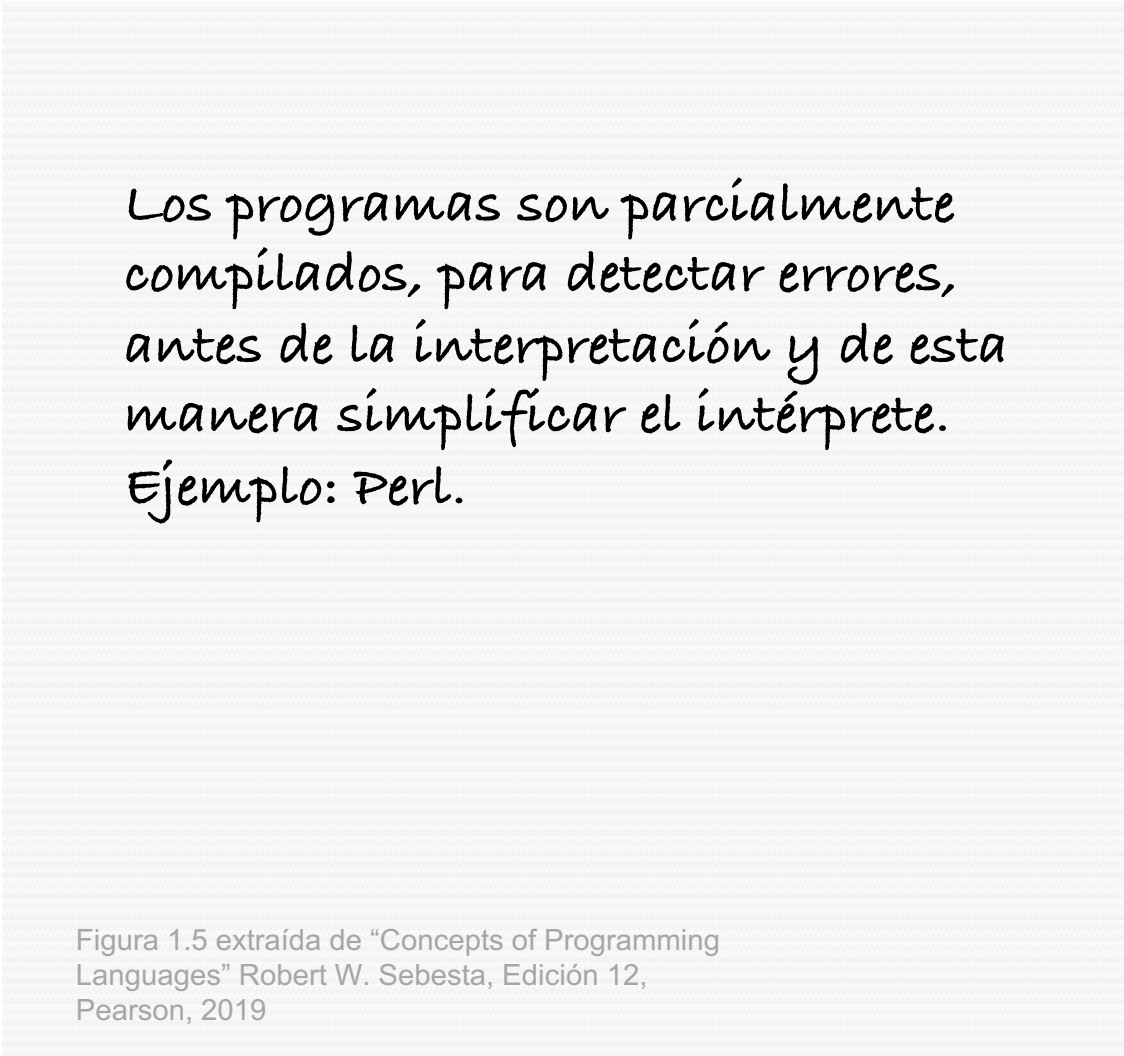
[illegible]

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

[illegible]

Sí hacemos una comparación entre compiladores e intérpretes se puede destacar que el programa que sea compilado será más rápido que uno interpretado, ya que en este caso la traducción se encuentra hecha y la computadora solo tiene que "leerla", mientras que el intérprete, al traducir código, solo cuando es necesario, se mantiene siempre en memoria y esto tiene un costo que justamente es el tiempo de ejecución del programa. Además del espacio de almacenamiento requerido.

[illegible]

Los programas son parcialmente compilados, para detectar errores, antes de la interpretación y de esta manera simplificar el intérprete.
Ejemplo: Perl.

Los programas son parcialmente compilados, para detectar errores, antes de la interpretación y de esta manera simplificar el intérprete.
Ejemplo: Perl.

[illegible]

- ❖ 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**.

A word cloud of programming languages. The words are arranged in a horizontal, somewhat circular pattern. The size of each word indicates its relative popularity or frequency. The most prominent words, shown in the largest font, are Java, Python, JavaScript, and PHP. Other visible words include C++, C#, Ruby, Perl, and Swift. Smaller words include Kotlin, Go, Rust, Julia, Haskell, F#, R, Scala, Clojure, Elixir, and many others. The background is a light, textured grey.

- ❖ 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.

Tiempos de ligadura



- ❖ 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.

[illegible]

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

A word cloud of programming languages. The largest word is 'C'. Other prominent words include 'Java', 'Python', 'JavaScript', 'PHP', 'C++', 'Perl', 'Ruby', 'Lisp', 'VB', 'Pascal', 'Assembly', 'Shell', 'Visual', 'C#', 'Objective-C', 'Fortran', 'Ada', 'BASIC', 'COBOL', 'Pascal', 'Lisp', 'VB', 'Pascal', 'Assembly', 'Shell', 'Visual', 'C#', 'Objective-C', 'Fortran', 'Ada', 'BASIC', 'COBOL'. The words are arranged in a circular pattern around the central 'C'.

- Desarrollado en clase.*

[illegible]

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

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

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

[illegible]

Imperar = mandar

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

- Programación estructurada
- Programación modular

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

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.

[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.

[illegible]

Ejemplos de lenguajes: Prolog.

¿Dudas?

