



Ciclo 1

Fundamentos de programación con Python

Sesión 7: Teoría de los lenguajes de programación 2

Programa Ciencias de la Computación e Inteligencia Artificial

Escuela de Ciencias Exactas e Ingeniería

Universidad Sergio Arboleda

Bogotá





Agenda

1. Clasificación de los lenguajes de programación
2. Lenguaje máquina
3. Lenguaje ensamblador
4. Lenguaje de alto nivel
5. Compilador
6. Intérprete
7. Paradigmas de programación





1. Clasificación de los lenguajes de programación

- Lenguajes de máquina o de bajo nivel.
- Lenguaje ensamblador o de intermedio nivel.
- Lenguajes de alto nivel.
- Lenguajes de muy alto nivel.
- Lenguajes naturales.





2. Lenguajes de máquina o de bajo nivel

- Es el único lenguaje que entiende el hardware y usa exclusivamente el sistema binario.
- Este lenguaje es específico para cada hardware
- El programa es almacenado en memoria.





3. Lenguaje ensamblador o de intermedio nivel.

- El lenguaje ensamblador surgió de la necesidad de desarrollar un lenguaje de nivel mayor, que fuese más comprensible que el de la máquina pero que permitiera acceder a los detalles de dicha máquina.
- Emplea mnemónicos (palabras cortas escritas con caracteres alfanuméricos), para codificar las operaciones.
- Los datos y direcciones son codificados generalmente como números en un sistema hexadecimal.





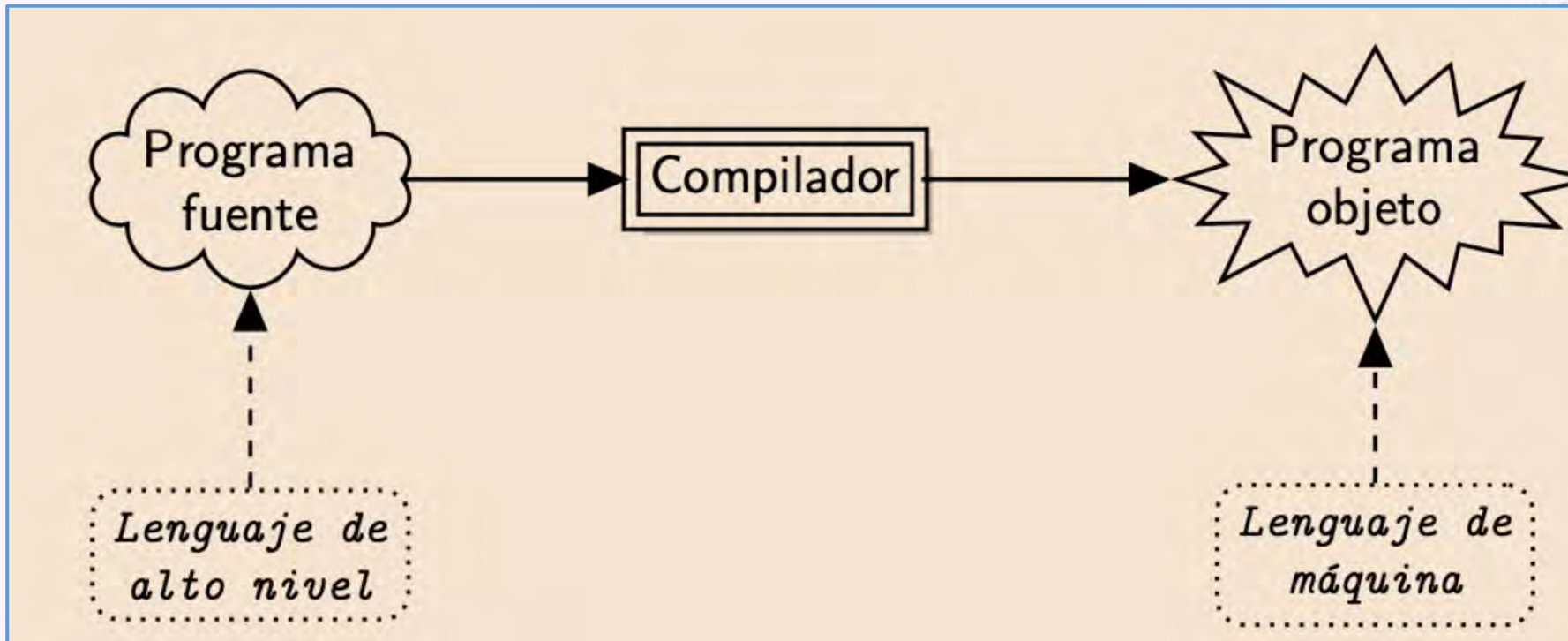
4. Lenguaje de alto nivel

- Estos lenguajes están basados en una estructura gramatical para codificar estructuras de control y/o instrucciones. Cuenta con un conjunto de palabras reservadas.
- Permiten el uso de símbolos aritméticos y relacionales para describir cálculos matemáticos, y generalmente representan las cantidades numéricas mediante sistema decimal.
- Permiten al programador olvidar el direccionamiento de memoria, ya que esto se realiza automáticamente por parte de un programa compilador o interprete.



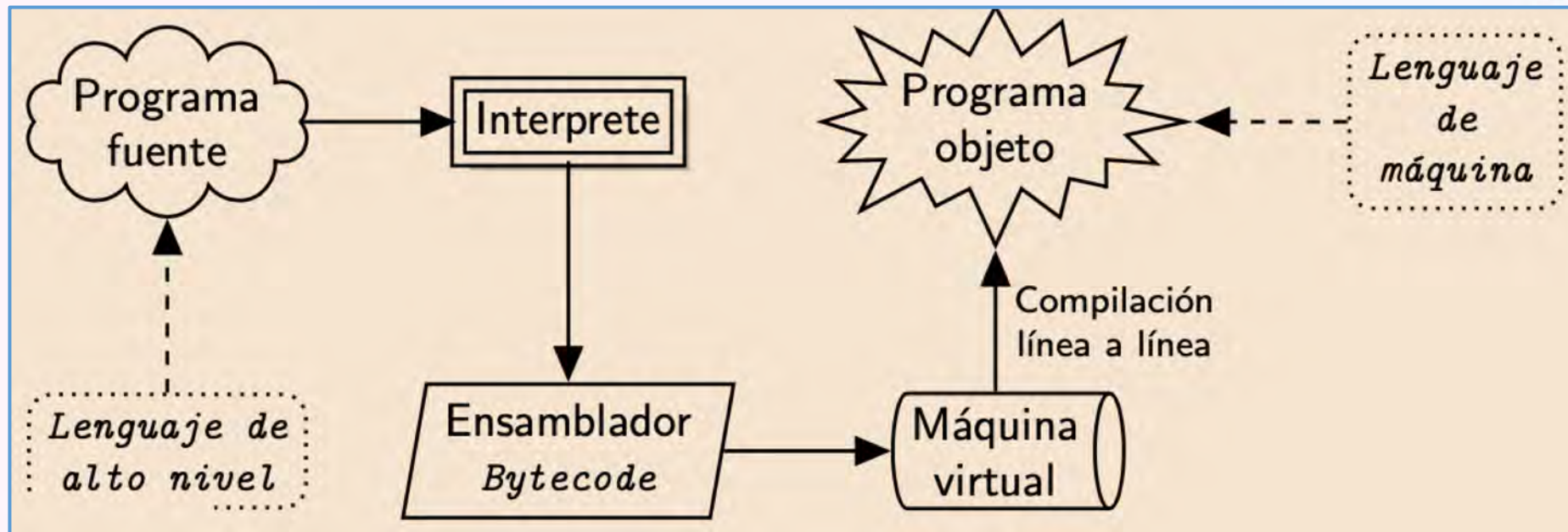


5. Compilador





6. Interprete





7. Paradigmas de programación

- Lenguajes de programación imperativos.
- Lenguajes de programación estructurados o procedurales.
- Lenguajes de programación declarativos.
- Lenguajes de programación orientados a objetos.





7.1 Imperativos

- Describe la programación en términos del estado de la memoria del programa y sentencias que cambian dicho estado.
- Los programas imperativos son un conjunto de instrucciones que se ejecutan secuencialmente y que le indican al computador cómo realizar una tarea.
- La implementación de hardware de la mayoría de computadores es imperativa ya que el hardware está diseñado para ejecutar código de máquina el cual es imperativo.
- **Ejemplos:** Ensamblador, Cobol, Pascal, Fortran, BASIC, C, Perl, Ada, MathLab, SciLab.





7.2 Estructurados

- En un lenguaje estructurado un programa se descompone en procedimientos individuales (funciones) que realizan una tarea específica, para resolver un problema en conjunto. Ejemplos: Pascal, C++, Visual Basic, Java, MathLab, SciLab.





7.3 Declarativos

- Basado en la utilización de predicados lógicos o funciones matemáticas.,
- El objetivo es conseguir lenguajes expresivos en los que no sea necesario especificar cómo resolver el problema, sino qué problema se desea resolver.
- Ejemplos: Lisp, ML, Haskell, Maude, Prolog, SQL.





7.4 Objetos

- Se definen una serie de conceptos y técnicas de programación para representar acciones o cosas de la vida real basada en objetos.
- Estos objetos se usan como instancias de clases y sus interacciones para diseñar aplicaciones y programas.
- Forma de operación: todo el programa está construido con base a diferentes componentes (objetos), cada uno tiene un rol específico en el programa y todos los componentes pueden comunicarse entre ellos de forma predefinidas.
- Para la construcción de las clases se utilizan técnicas tales como: abstracción, herencia, modularidad, polimorfismo, encapsulamiento y ocultamiento.
- **Ejemplos:** Smalltalk, C++, Java, Python, R.





Preguntas

