

ANÁLISIS COMPARATIVO DE LENGUAJES

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



Bibliografía

- “Programming Languages. Design and Implementation” Terrence W. Pratt y Marvin V. Zelkowitz, Edición 4, Prentice Hall, 2001.
- “Concepts of Programming Languages” Robert W. Sebesta, Edición 12, Pearson, 2019.
- Apuntes de Cátedra.



Diseño de Lenguajes de Programación

Notas de Clase I - Año 2021

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

Capítulos I y II Concepts of Programming Languages - Robert Sebesta

Objetivos Generales de la Asignatura



- Estudiar las características principales del diseño de Lenguajes de Programación, para ello estudiaremos:
 - Características que influyen en el diseño de Lenguajes de Programación.
 - Criterios de evaluación - atributos de un buen lenguaje.
 - Lenguaje Prolog.
 - Descripción formal de los lenguajes.
 - Tipos de datos - Objetos de datos.
 - Administración de la memoria.
 - Control de datos entre subprogramas.
 - Variantes en el control de subprogramas.
 - Características principales de la POO.



¿Qué pensamos que es un lenguaje de programación?

- *Lluvia de ideas:*

Lenguajes de Programación

- Podemos decir que es cualquier notación formal para describir algoritmos y estructuras de datos, con la intención que pueda ser entendida por una computadora.
- Lenguajes formales que, mediante una serie de instrucciones, le permite al programador escribir un conjunto de órdenes, acciones, datos y algoritmos, para de esta forma crear programas que controlen el comportamiento físico y lógico de una máquina.
- Etc.

Dominios de Programación



El diseño y evaluación de un lenguaje particular es altamente dependiente del dominio en el cual es usado:

- ❖ **Aplicaciones Científicas** - Gran cantidad de computaciones con números de punto flotante; uso de arreglos

Fortran, ALGOL 60

- ❖ **Aplicaciones de Negocios** - Producir reportes, usar números decimales y caracteres

COBOL, RPG

- ❖ **Inteligencia Artificial** - Símbolos más que números son manipulados; uso de listas vinculadas

LISP, Prolog, Scheme, Python

- ❖ **Web Software** - Lenguajes desde markup (HTML) a scripting para propósito general

Java, PHP, JavaScript

Criterios de Evaluación de Lenguajes

- ❖ **Readability (Legibilidad)**: facilidad con la cual los programas pueden ser leídos y entendidos.
- ❖ **Writability (Facilidad de Escritura)**: facilidad con la cual un lenguaje puede ser usado para crear programas en el dominio de programas.
- ❖ **Reliability (Confiabilidad)**: conformidad con las especificaciones, es decir, llevarlas a cabo bajo todas las condiciones.
- ❖ **Costo**: costo total definitivo.
- ❖ **Otros**: portabilidad, generalidad, bien definido.

Criterios de Evaluación: Readability

Facilidad con la cual los programas pueden ser leídos y entendidos



□ Tipos de Datos y estructuras

- ❖ Adecuados tipos de datos predefinidos
Ejemplo: tipos numéricos vs. tipos Boolean para variables
- ❖ Estructuras adecuadas, tales como arreglos.
- ❖ Estructuras de datos definidas por el programador, tales como registros.

□ Consideraciones de Sintaxis

- ❖ Formas de los identificadores: composición flexible
- ❖ Palabras especiales

Criterios de Evaluación: Writability

Facilidad con la cual un lenguaje se puede usar para crear programas



- **Difiere de acuerdo al dominio para el cual se diseña el lenguaje.**
- **Simplicidad y ortogonalidad**
 - ❖ Es mejor tener un número pequeño de construcciones bien definidas, un número pequeño de primitivas, y un conjunto consistente de reglas para combinarlas.
- **Expresividad**
 - ❖ Un conjunto de construcciones relativamente convenientes de especificar operaciones
 - ❖ Count ++
 - ❖ Uso de For para loops

Criterios de Evaluación: Confiabilidad



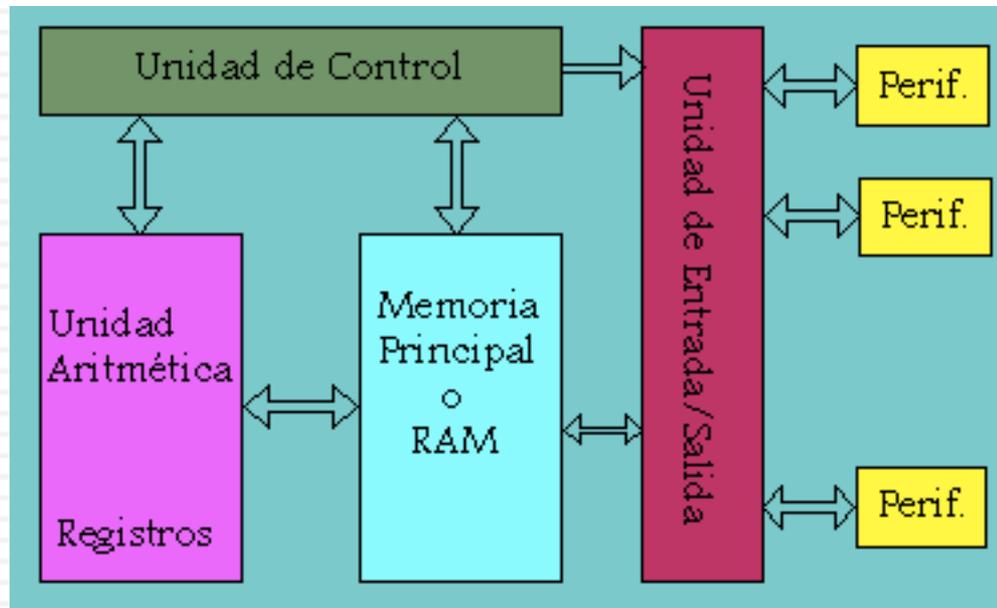
- *Chequeo de tipos*
- *Manejo de excepciones*
- *Alias restringidos*
- *Legibilidad y Facilidad de escritura*

Aspectos a considerar para el diseño de lenguajes de programación

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

1- La computadora subyacente en donde se van a ejecutar los programas escritos en el lenguaje.

El modelo de arquitectura subyacente para lenguajes procedurales se conoce como arquitectura de von Neumann.



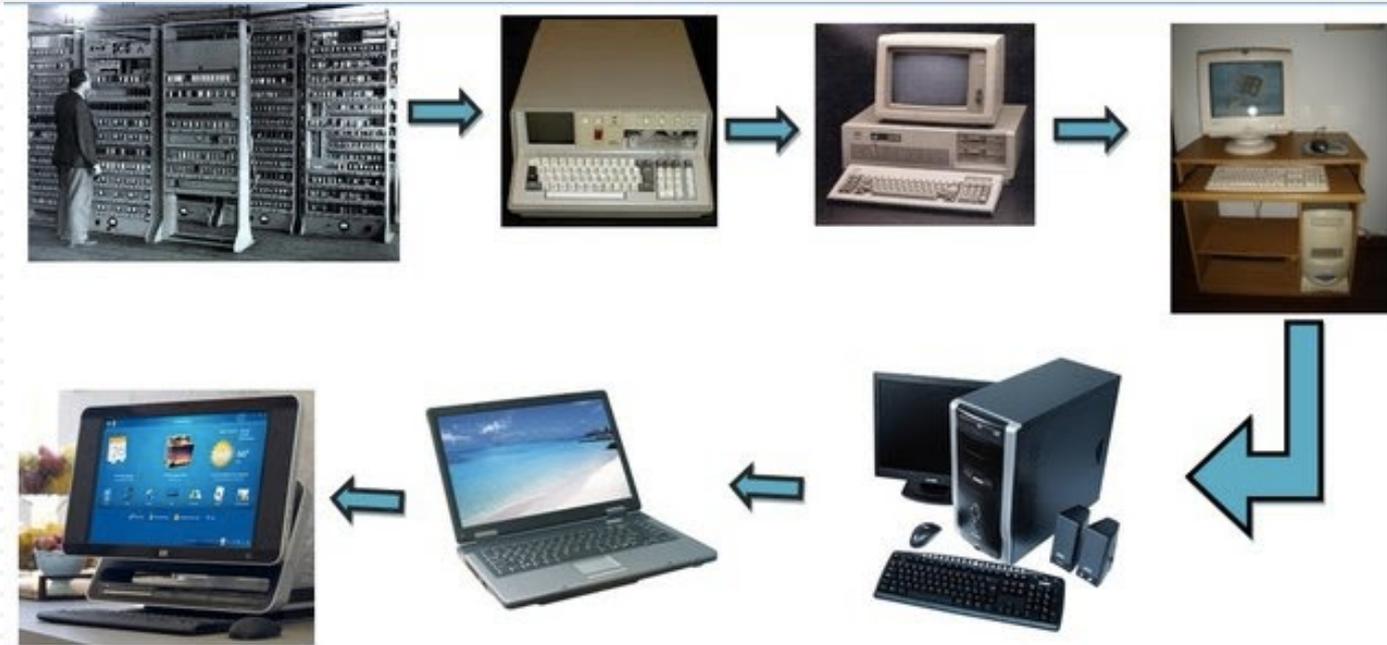
Componentes fundamentales de una computadora

- ❖ Datos
- ❖ Operaciones Primitivas
- ❖ Control de Secuencia
- ❖ Acceso a Datos
- ❖ Gestión de Almacenamiento
- ❖ Entorno de Operación

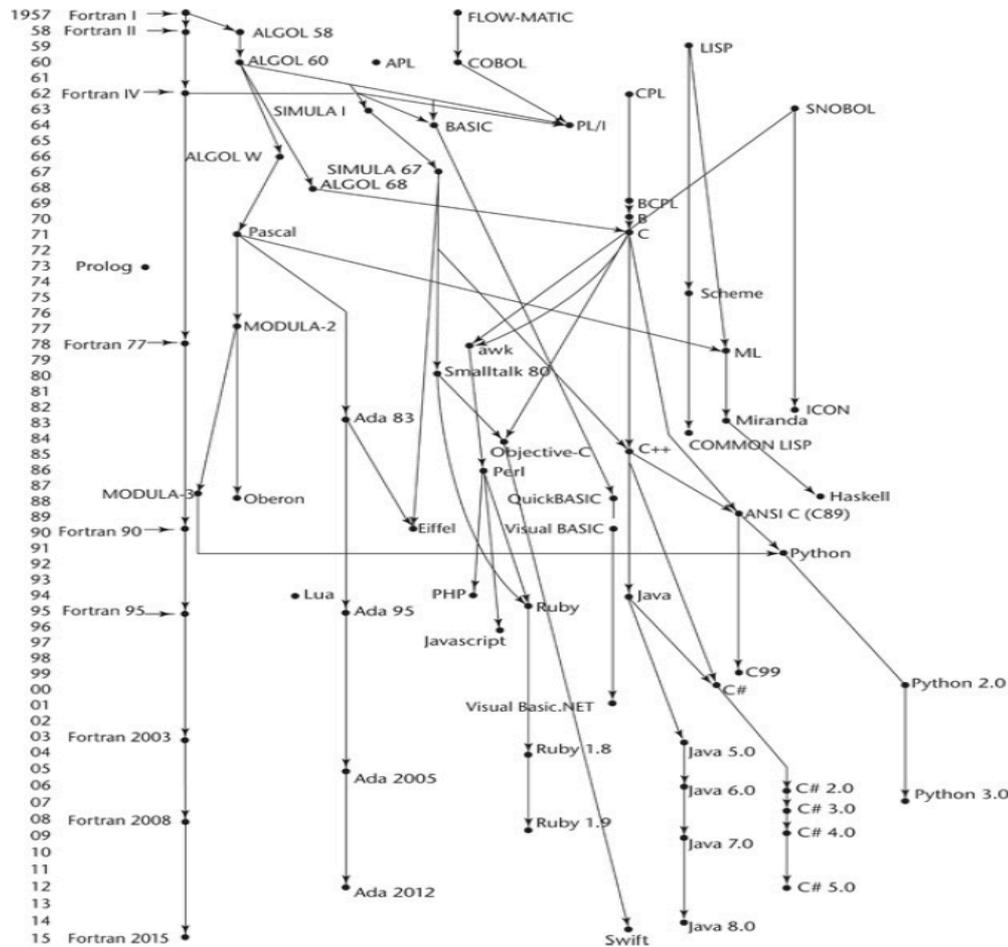


¿Qué relación se puede establecer con los lenguajes de programación?

Un poco de historia



Evolución de los lenguajes de programación



¿Dudas?

