UNIDAD 1: INTRODUCCION

CONTENIDO

1. LENGUAJES DE PROGRAMACIÓN
   CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN
   EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN
2. SISTEMAS FORMALES
   TEORÍA DE LENGUAJES FORMALES Y AUTÓMATAS
   APLICACIÓN DIRECTA DE CONCEPTOS PROPIOS DE LAS CIENCIAS DE LA COMPUTACIÓN
3. TRADUCTORES
   COMPILADOR
   INTERPRETE

LENGUAJES DE PROGRAMACIÓN

LENGUAJES DE PROGRAMACIÓN

Lenguaje natural:

- Las relaciones humanas se llevan a cabo a través del lenguaje. Una lengua permite la expresión de ideas y de razonamientos, y sin ella la comunicación sería imposible.

Traductor

Lenguaje de Programación:

- Notación que se usa para describir Algoritmos y Estructuras de Datos y que se puede implementar en una Computadora.
- La programación de ordenadores se realiza en los llamados *lenguajes de programación* que posibilitan la comunicación de órdenes a la computadora u ordenador.

Traductor: Compilador o Interprete

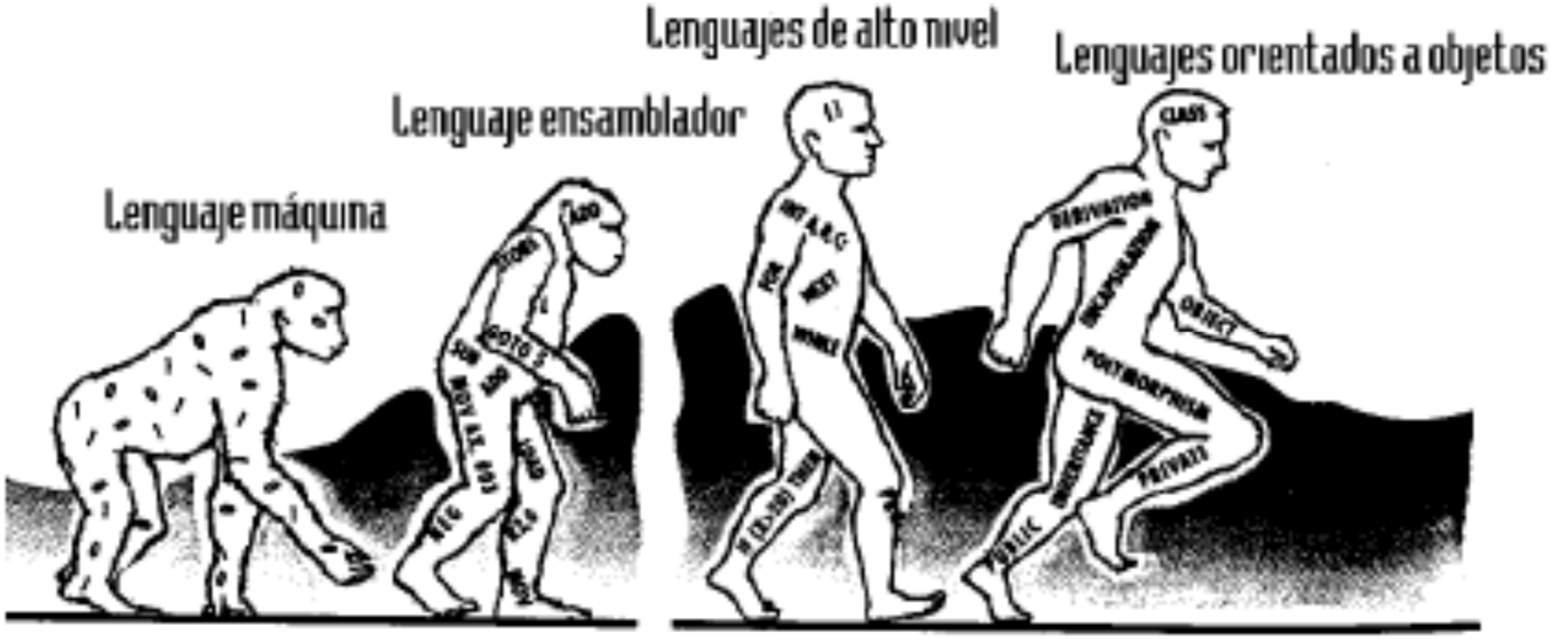
LENGUAJES DE PROGRAMACIÓN

- Un **lenguaje de programación** se puede definir:

  - *Una notación formal para describir algoritmos o funciones que serán ejecutadas por un ordenador.*
  - *Es un lenguaje para de comunicar instrucciones al ordenador.*
  - *Es una convención para escribir descripciones que puedan ser evaluadas.*
  - *Conjunto de normas lingüísticas (notación) que permiten escribir un programa con instrucciones que sean entendidas, directa o indirecta, por la máquina.*

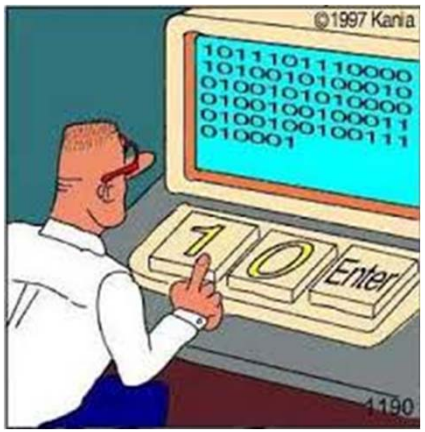
CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

- *Según el grado de independencia de la máquina*
  - *Lenguaje máquina*
  - *Lenguaje ensamblador* (en inglés *assembly*)
  - *Lenguajes de medio nivel*
  - *Lenguajes de alto nivel*


Lenguaje máquina
Lenguaje ensamblador
Lenguajes de alto nivel
Lenguajes orientados a objetos

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN: Lenguaje de máquina

- Instrucciones directamente entendibles por el computador sin la necesidad de traducción alguna.
- Instrucciones: secuencia de **0s** y **1s** (*bits*) que especifican operación a realizar, registros del procesador, celdas de memoria, etc.
- Dependientes de la máquina.
- Fáciles de comprender para un computador, difíciles para el hombre.



CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN: Lenguaje ensamblador

- Las instrucciones se escriben en códigos alfabéticos: **mnemotécnicos.**
- Son generalmente dependientes de la máquina.
- Los mnemotécnicas son más fáciles de recordar. Una instrucción típica de ensamblador puede ser: ADD x,y,z
- Requiere de un proceso de traducción a lenguaje de máquina.
- El traductor de programas en lenguaje ensamblador a lenguaje de maquina se le llama **ensamblador.**

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN: Lenguajes de medio nivel

- Tienen algunas de las características de los lenguajes de ensambladores o bajo nivel (posibilidad de acceso directo a posiciones de memoria, y hardware en general, etc...) más el manejo de estructuras de control y de datos de los lenguajes de alto nivel.
- Más amigable que los lenguajes anteriores.
- Ejemplos de este tipo de lenguajes son el **C** y el FORTH.
- El proceso de Traducción es complejo.

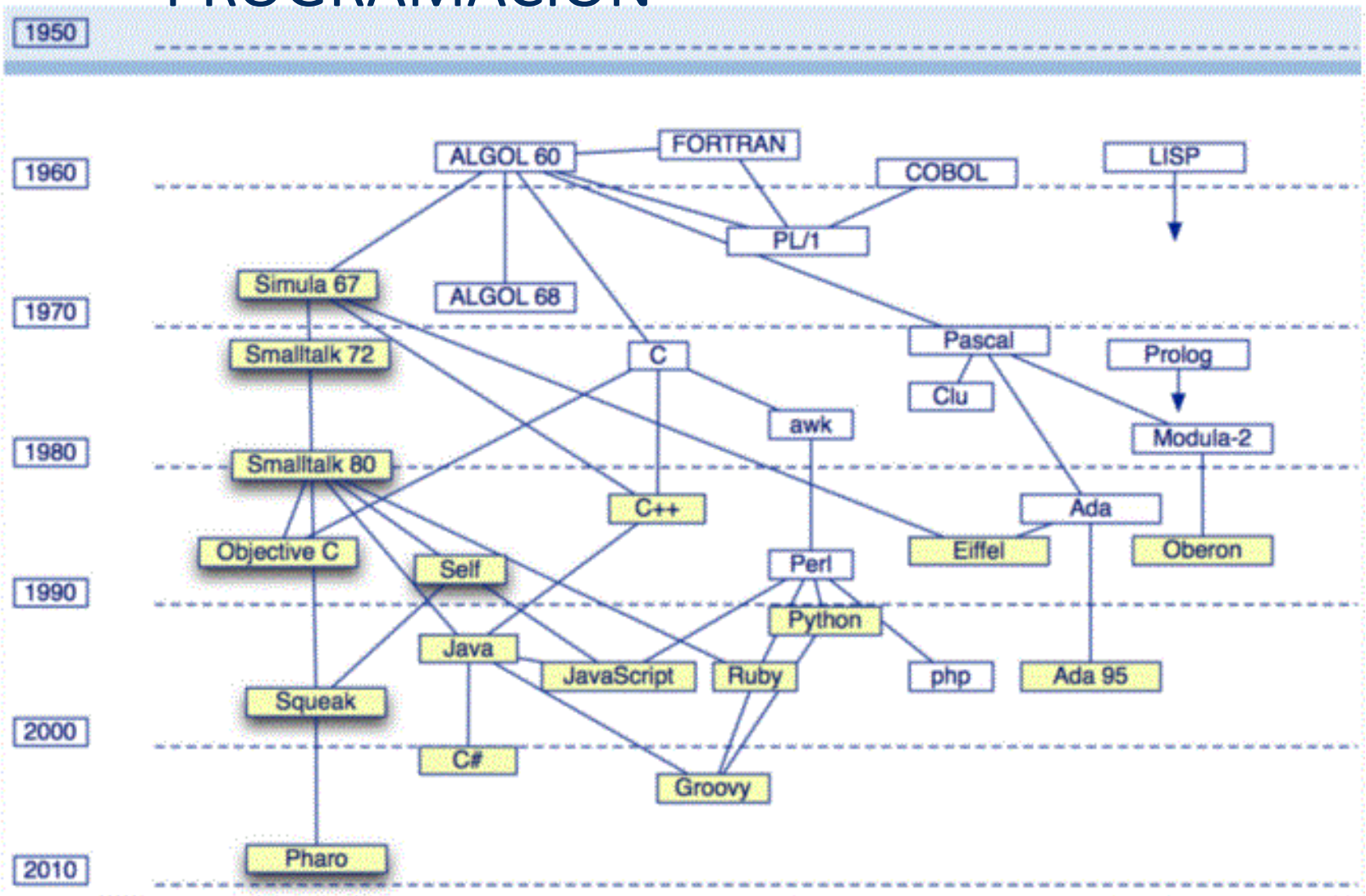
CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN: Lenguajes de alto nivel

- Instrucciones escritas con palabras similares a las de los lenguajes humanos (en la mayoría de los casos, el Inglés
- Fácil escritura y comprensión del código para el programador.
- Existen muchos lenguajes de alto nivel, por citar algunos: ADA, BASIC, COBOL, FORTRAN, C , C++, Modula-2, Lisp, Prolog, Pascal, Java, Php, html, xml , etc.
- El proceso de Traducción es más complejo

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

- *Según la forma de sus instrucciones*
  - *Lenguajes imperativos o procedimentales*
  - *Lenguajes declarativos: lógicos y funcionales*
  - *Lenguajes concurrentes*
  - *Lenguajes orientados a objetos*
- *Por generaciones*
  - *Primera generación*
  - *Segunda generación*
  - *Tercera generación*
  - *Cuarta generación*
  - *Quinta generación*
  - *Generación orientada a objetos*
  - *Generación visual*
  - *Generación internet*

EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN



https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoria/index.html

SISTEMAS FORMALES

SISTEMAS FORMALES: TEORÍA DE LENGUAJES FORMALES Y AUTÓMATAS

- Consisten en un conjunto de palabras llamadas axiomas y un conjunto finito de relaciones, llamadas reglas de producción o inferencias
- Ejemplos de Sistemas Formales: la Teoría de conjuntos, el álgebra de Boole y la Forma Normal de Backus, **Teoría de Lenguajes Formales, Gramáticas y Autómatas**
- Son importantes para el diseño, la implementación y el estudio de los lenguajes de programación. Se utilizan en particular para especificar la sintaxis y la semántica de los lenguajes de programación.

APLICACIÓN DIRECTA DE CONCEPTOS PROPIOS DE LAS CIENCIAS DE LA COMPUTACIÓN

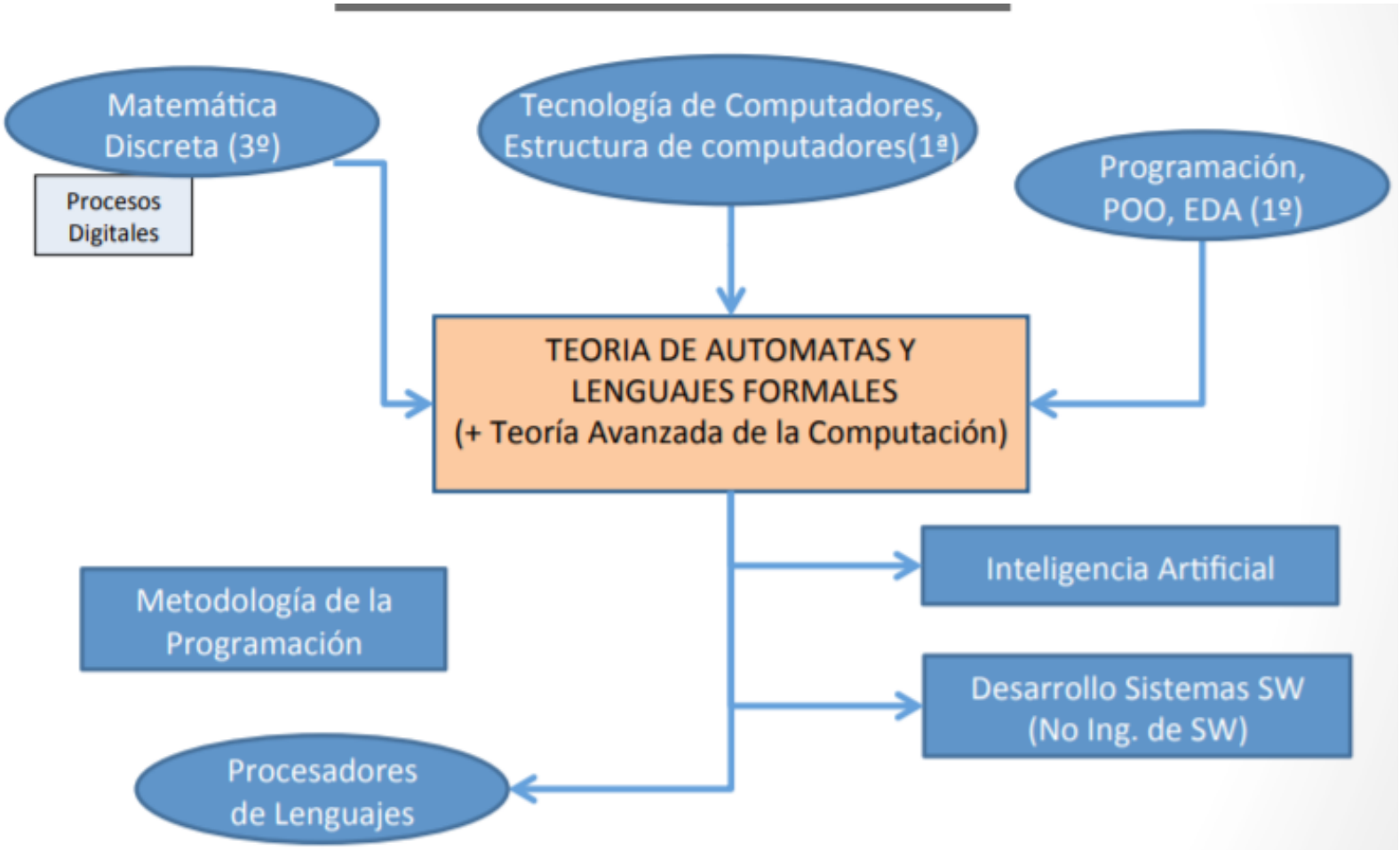
- Videojuegos
  - Comportamiento de personajes
- Compiladores y Procesamiento de Lenguaje Natural
  - Análisis Léxico en lenguajes programación (compilador).
  - Búsqueda de cadenas o comparación de “patrones”
  - Diseño de nuevos lenguajes de programación o ampliación
- Implementación de Protocolos Robustos
  - Para clientes o usuarios
  - E.g. Sistemas de Seguridad

APLICACIÓN DIRECTA DE CONCEPTOS PROPIOS DE LAS CIENCIAS DE LA COMPUTACIÓN

- Diseño de estructuras y “parsing”: gramáticas (ej: XML)
  - Búsqueda de cadenas o comparación de “patrones”
- SW para diseñar y evaluar circuitos digitales.
- “Escanear” grandes cantidades de texto (web)
- SW para verificar sistemas que tienen un número finito de “estados”

http://ocw.uc3m.es/ingenieria-informatica/teoria-de-automatas-y-lenguajes-formales/material-de-clase-1/tema-i-introduccion/view

RELACIÓN CON OTRAS ÁREAS.



TRADUCTORES

TRADUCTORES: COMPILADORES e INTERPRETES

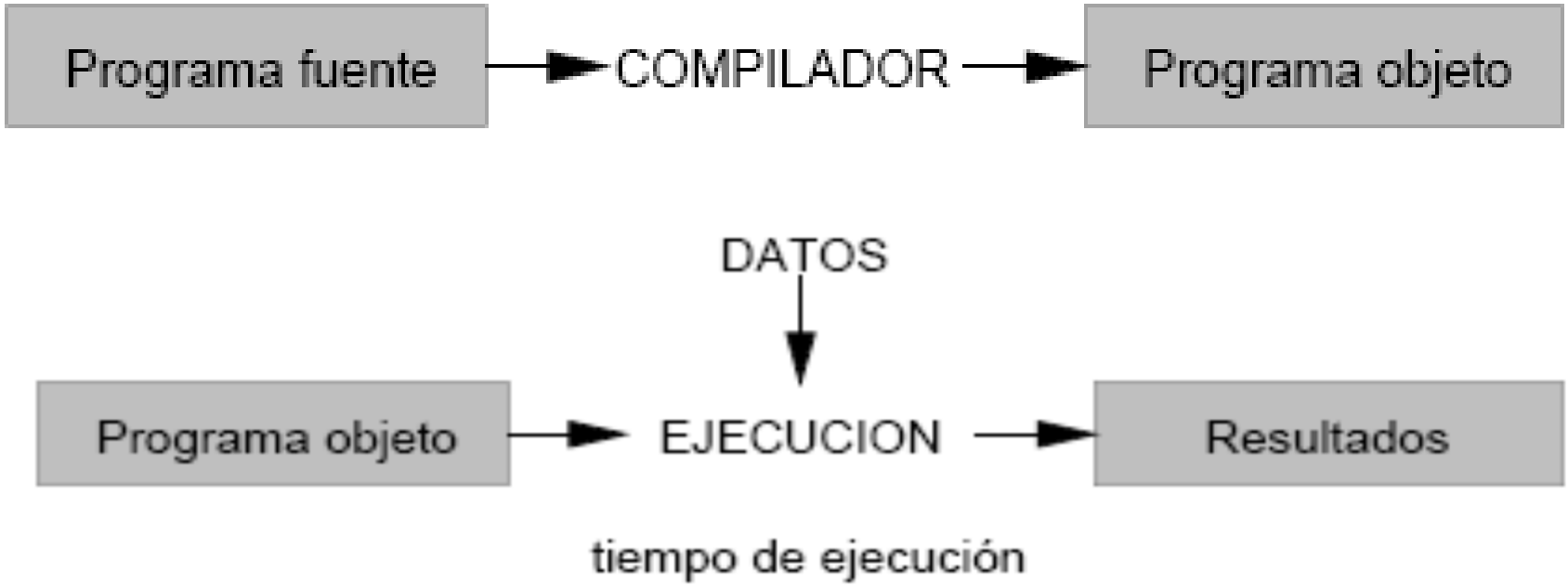
- El programa original se denomina *programa fuente* y el programa traducido en lenguaje máquina se llama *programa objeto*. El traductor de programas fuente a objeto es un programa llamado **compilador o intérprete.**

TRADUCTORES: COMPILADORES e INTERPRETES

- Un lenguaje de programación se implementa construyendo un traductor, el cual traduce programas que están escritos en algún lenguaje de programación a lenguaje de maquina para ser ejecutados directamente por una computadora.

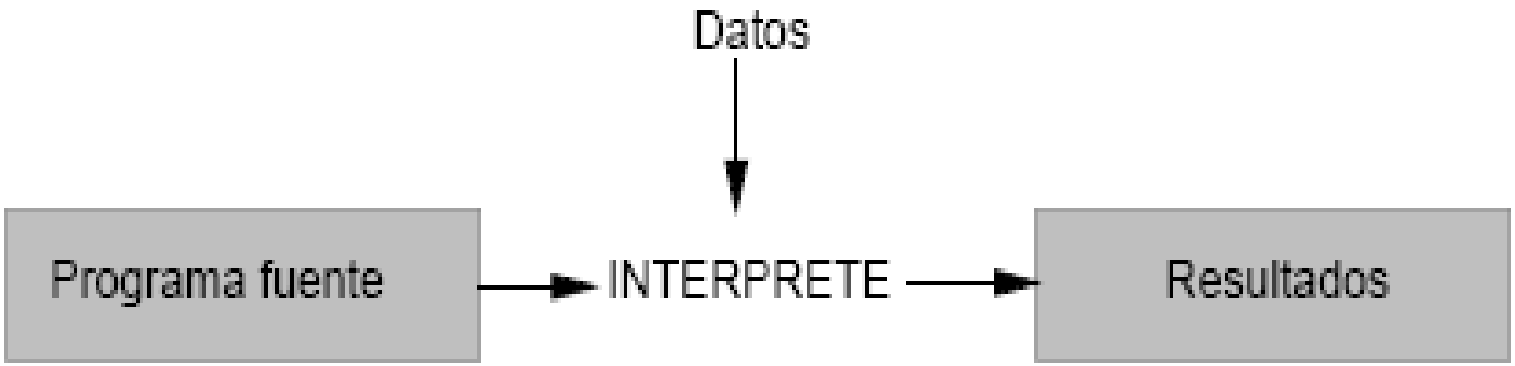
COMPILACIÓN

- Proceso de traducción que convierte un programa fuente escrito en un lenguaje de alto nivel a un programa objeto en código maquina y listo por tanto para ejecutarse en un ordenador, con poca o ninguna preparación adicional. (Ver Figura)



INTÉRPRETE

- Proceso de traducción paso a paso, conforme va ejecutándose el programa fuente.
- Un interprete puro ejecuta directamente el programa fuente, coexistiendo en la memoria del ordenador el programa fuente y el intérprete (Ver Figura)
- Antiguamente el uso de intérpretes era más barato por el espacio de memoria menor frente al uso de compiladores.



BIBLIOGRAFIA

- http://ocw.uc3m.es/ingenieria-informatica/teoria-de-automatas-y-lenguajes-formales/material-de-clase-1/tema-i-introduccion/view
- E. Alfonseca Cubero, M. Alfonseca Moreno, R. Moriyón Salomón. Teoría de Autómatas y Lenguajes Formales. Ed. McGraw-Hill, 2007