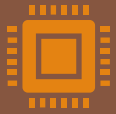


# PROGRAMACIÓN ESTRUCTURADA

UNIDAD 1



Es una técnica para construir transformar la solución a un problema o necesidad en un software.

Se utilizan diagramas de flujo para modelar la solución.  
Se utilizan lenguajes de programación para transformar esa solución en un software.



Existen diferentes lenguajes de programación para este fin

# ¿Qué es la programación?

# Algoritmos, diagramas de flujo

# ¿Qué es un algoritmo?

---

Es una metodología para resolver un problema

## Características de los algoritmos

- Definido:
  - Deben conocerse todos los pasos para resolver el problema
- Preciso
  - Debe ser claro y puntual.
- Finito
  - Debe tener un comienzo y un fin

# Algoritmos

---

Existen algoritmos numéricos y no numéricos.

Hay diferentes formas de representar un algoritmo

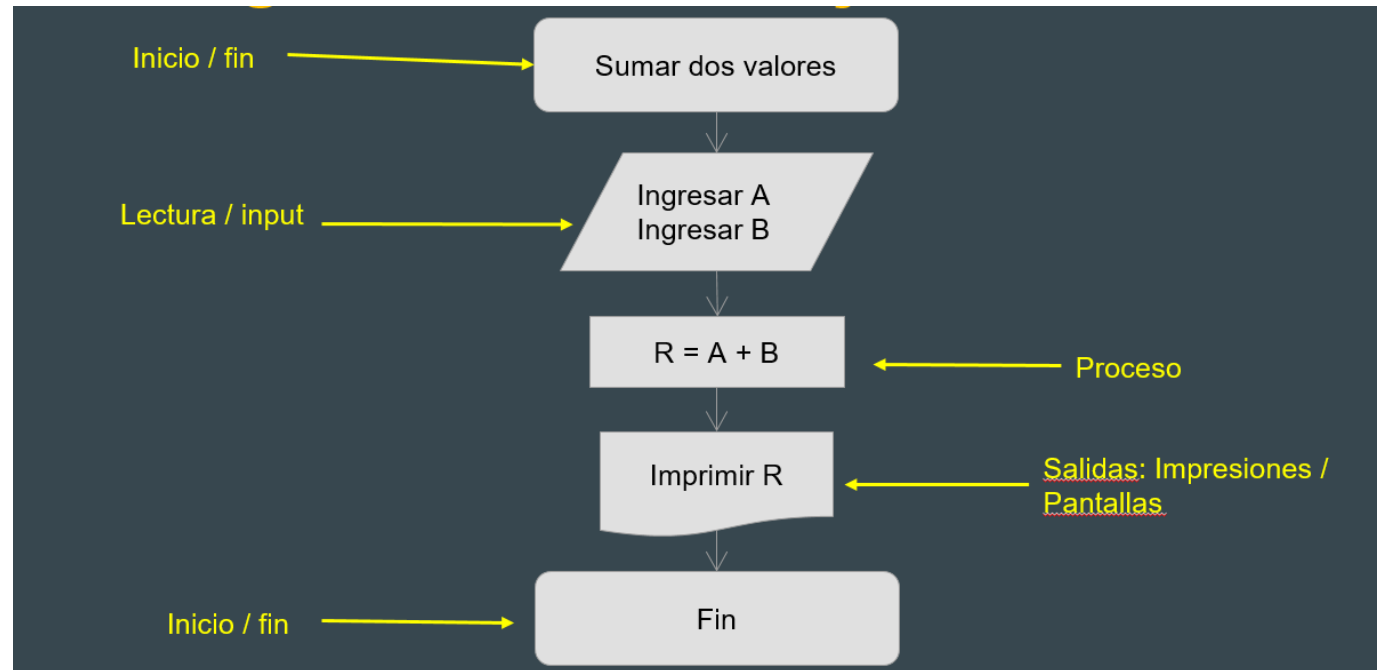
- Gráfico
  - Utilizando diagramas de flujo
- No gráfico
  - Utilizando pseudocódigo

Hay diferentes operadores que se utilizan en la construcción de algoritmos

- Aritméticos (Suma, resta, multiplicación, división, etc).
- De comparación ( $>$ ,  $<$ ,  $=$ , etc).
- Lógicos (And, or, not, etc).

# Algoritmo gráfico: diagramas de flujo

- Método gráfico para resolver un algoritmo.
- Utiliza símbolos que representan, entradas, procesos, salidas, deci



# Algoritmo vs. Programa

---

De fácil comprensión para las personas, no para las computadoras.

Lenguajes expresivamente muy ricos (ambigüedad)

Los lenguajes de programación tienen una sintaxis rígida y una semántica definida

# ¿Qué es un paradigma?

---

De fácil comprensión para las personas, no para las computadoras.

Lenguajes expresivamente muy ricos (ambigüedad)

Los lenguajes de programación tienen una sintaxis rígida y una semántica definida



# Historia de los lenguajes de programación

LOS PRIMEROS LENGUAJES ERAN SIMPLES CÓDIGOS Y SE PUEDEN VER EN TARJETAS PERFORADAS DE MAQUINAS DE TEJER EN EL 1801, CONOCIDA COMO MAQUINA DE JACQUARD

# Máquina Analítica

---

En 1842 Lovelace tradujo varias memorias matemáticas y generó esta máquina que resolvía el cálculo de los números de Bernoulli

Números de Bernoulli, es una secuencia de números racionales.

Este se considera el primer programa

# ¿Cuál fue el primer lenguaje de programación?

---

Es difícil decir cual es

Se dependía muchísimo del hardware y sus limitaciones

Surgen a partir de 1940 una serie de lenguajes matemáticos

En 1955 surge FORTRAN

En 1958 surge LISP

En 1959 surge COBOL

# Siglo XX

---

Las primeros programas eran específicos para una determinada función, trenes, cálculos numéricos

Al principio eran solo cálculos , luego se comenzó a poder representar la “lógica”

En los 80 cambia la idea, dejan de desarrollarse nuevos paradigmas para buscar una interrelación entre los ya existentes y se busca que los lenguajes puedan responder a mas de uno de ellos

En los 90 hay un rápido crecimiento de internet revolucionando a los lenguajes y ofreciéndoles nuevos desafíos y adaptaciones, el mas popular fue Java

Muchos de estos lenguajes venían con su IDE

# 1960-1980

---

Surgen cientos de lenguajes que responden a distintos paradigmas

Que es un paradigma ?

- Estructurado.....C
- Lógico .....Prolog
- Funcional .....Goofer, Miranda
- Orientado a objetos .....Smalltalk, Simula

A partir de allí descendieron muchos lenguajes que se encuadraban dentro de estos paradigmas

# Tendencias actuales

---

Aumentar el soporte para la programación funcional

Construir lenguajes para apoyar la programación concurrente y distribuida

Lenguajes mas seguros y confiables

Distribución y movilidad

Mayor integración a bases de datos

Código abierto

Programación orientada a Aspectos

# CLASIFICACION DE LOS LENGUAJES DE PROGRAMACION SEGÚN SU NIVEL

LOS LENGUAJES DE PROGRAMACIÓN SON AQUELLOS QUE NOS PERMITE COMUNICARNOS CON LAS COMPUTADORAS. SON SÍMBOLOS Y PALABRAS QUE LE PERMITEN AL USUARIO DARLE ORDENES A LA COMPUTADORA PARA QUE LAS EJECUTE,

# LENGUAJE DE MAQUINA

---

Utiliza el alfabeto binario ..... 0, 1

Estos 0, 1 son conocidos como bits

Con ellos se forman cadenas binarias que escriben las instrucciones que el microprocesador entedera y ejecutara

Este lenguaje fue el inicial , se dejo de usar por su dificultad y facilidad de cometer errores para escribirlo



# LENGUAJES DE BAJO NIVEL

---

De mas fácil escritura

Depende de la maquina donde se implementara

El assembler fue el primer lenguaje de estos

Se genera un lenguaje fuente, se traduce a lenguaje de maquina y se genera un lenguaje objeto

Pueden generar programas rapidos

Desventajas: son mas difíciles de aprender, específicos para un procesador

# LENGUAJES DE ALTO NIVEL

---

Son independientes de la maquina adonde se implementara

Es fácil cambiar de una maquina a otra

Necesitan de un compilador o interprete para traducirlo a lenguaje de maquina

Imperativos .....cobol, C , Pascal

Logicos .....prolog

POO .....Smalltalk, C++, Java, .net

Orientados al problema ....aquellos específicos para gestión

# LENGUAJES DE ALTO NIVEL

---

De 1ra generación.....lenguaje de maquina, assembler

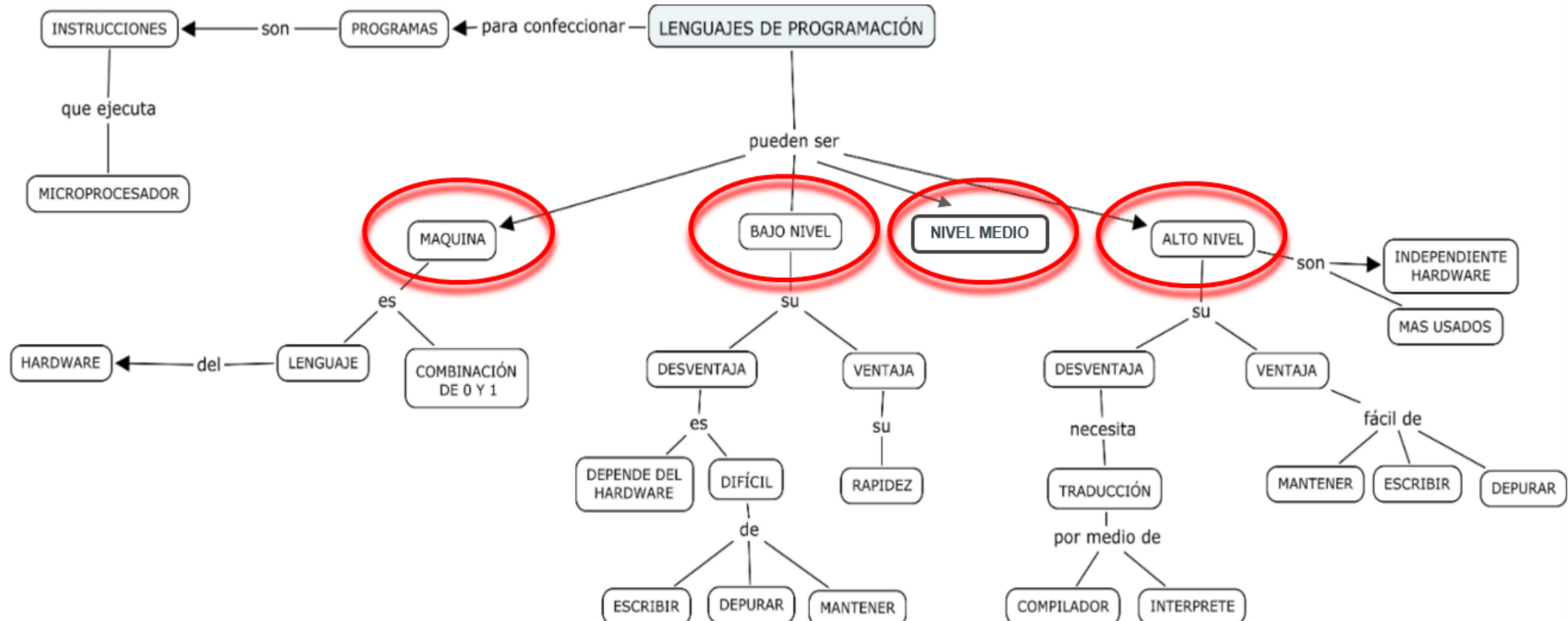
De 2da generación.....Cobol, Fortran

De 3ra generación..... C++, Pascal

De 4ta generación.....de gestión , SQL

De 5ta generación..... Inteligencia artificial , Prolog

# Lenguajes de programación



# ENSAMBLADORES, COMPILADORES E INTERPRETES

EJECUTANDO UN PROGRAMA

# Ensamblador

---

Procesan enunciados del código fuente en lenguaje ensamblador y los traducen en lenguaje máquina para ser ejecutados por un procesador.

Permiten que los programas se escriban en una computadora para generar un archivo que pueda ejecutarse en cualquier otra.

El ejecutable en lenguaje objeto se carga y ejecuta en el sistema destino.

# Compilador

---

## Compilador

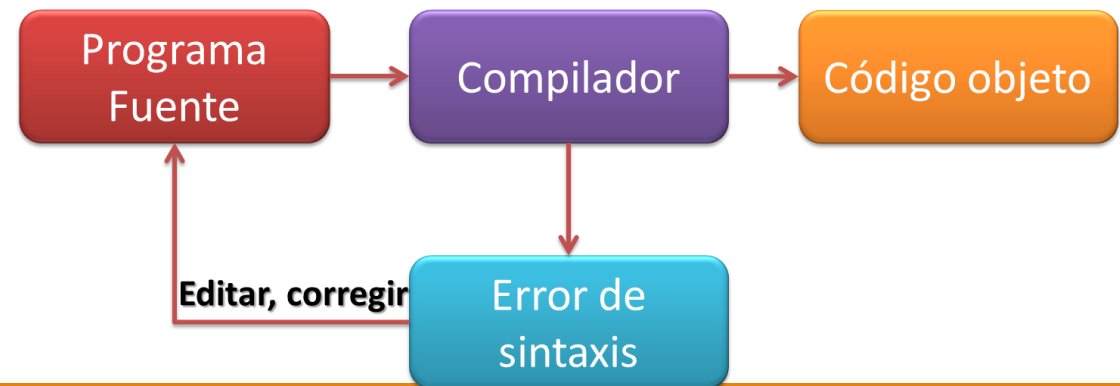
Transforma código fuente en un lenguaje de alto nivel a un lenguaje de bajo nivel.

Opera en fases (tokens). Cada fase, lleva a cabo una tarea sobre el código fuente,

Son dependientes de la plataforma (un programa compilado para linux no corre en windows)

## Lenguajes compilados

- Pascal
- C / C++
- Cobol
- .NET



# Intérprete

Programa que es capaz de analizar y ejecutar otros programas escritos en lenguaje de alto nivel

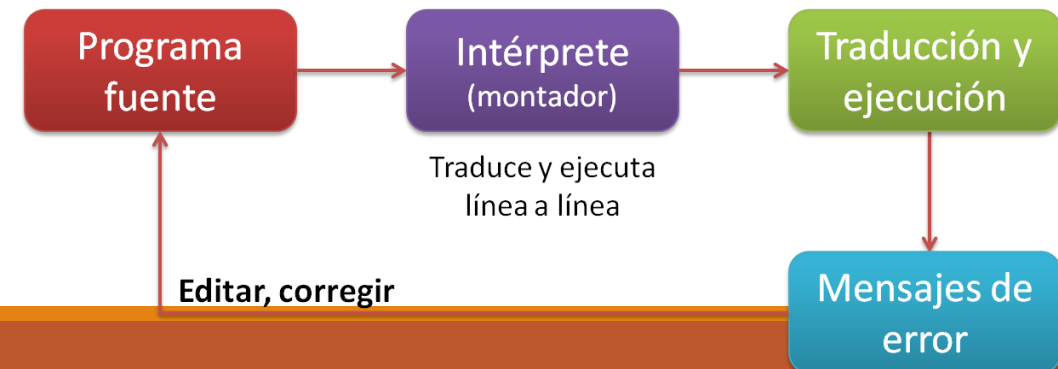
---

Realiza la traducción del programa fuente a código de máquina a medida que se lo necesita, normalmente instrucción por instrucción. Normalmente, no se guarda el resultado de la traducción. Son independientes de la plataforma

La ejecución es menos eficiente. Cada instrucción debe pasar por interpretación en tiempo de ejecución

Lenguajes Interpretados

- Ruby, Python, JavaScript





# FASES EN LA CONSTRUCCION DE UN PROGRAMA

LA IMPORTANCIA DE ESTAR ORGANIZADOS

# Fases en la construcción de un programa

---

## Análisis del problema

- El problema se analiza teniendo presente la especificación de los requisitos dados por el cliente de la empresa o por otra persona que encarga el programa.

## Diseño del algoritmo

- Una vez analizado el problema, se diseña una solución que conducirá a un algoritmo que resuelva el problema.

## Codificación

- La solución se escribe en la sintaxis del lenguaje de alto nivel (por ejemplo, C ) y se obtiene un programa.

# Fases en la construcción de un programa

---

## Ejecución, verificación y depuración

- El programa se ejecuta, se comprueba rigurosamente y se elimina todos los errores (denominados “bugs”, en inglés) que puedan aparecer

## Mantenimiento

- El programa se actualiza y modifica, cada vez que sea necesario, de modo que se cumplan todas las necesidades de cambio de sus usuarios.

## Documentación

- Escritura de las diferentes fases del ciclo de vida del software, esencialmente el análisis, diseño y codificación, unidos a manuales de usuario y referencia, así como normas para el mantenimiento

# Ciclo de vida de un sistema

---

Comienza desde que se recibe los requerimientos por parte del cliente hasta que el mismo se vuelve obsoleto, ya sea por cambios en el negocio, avances tecnológicos o dejar de usarlos