

Programación Estructurada

INTRODUCCIÓN A LA PROGRAMACIÓN



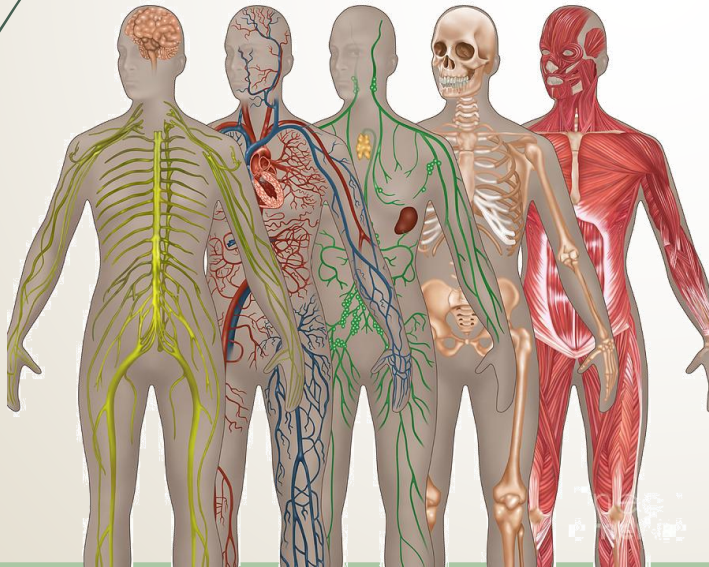
Índice

- Concepto de sistema
- Sistemas de procesamiento de información
- Concepto de computadora
- Concepto de Algoritmo
- Metodología
- Soluciones basadas en computadora
 - Análisis del problema, Diseño, Codificación, Compilación y ejecución, Verificación y depuración, Documentación y Mantenimiento
- Conceptos generales

Concepto de Sistema

- Un sistema es un conjunto de elementos interconectados que trabajan de forma coordinada para realizar alguna tarea compleja.

Por ejemplo: sistema circulatorio humano, sistema de tráfico aéreo, sistema solar, etc.



Procesamiento de Información (1)

- Un sistema de procesamiento de información es un sistema que transforma datos brutos en información organizada, significativa y útil.



DATOS BRUTOS

PROCESAMIENTO

INFORMACIÓN



Procesamiento de Información (2)

► Componentes:

- Entrada (datos),
- Procesador (métodos de transformación)
- Salida (información procesada)



Computadoras (1)

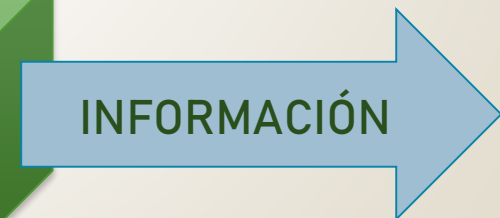
- Dispositivo electrónico capaz de procesar información.



PROCESADOR

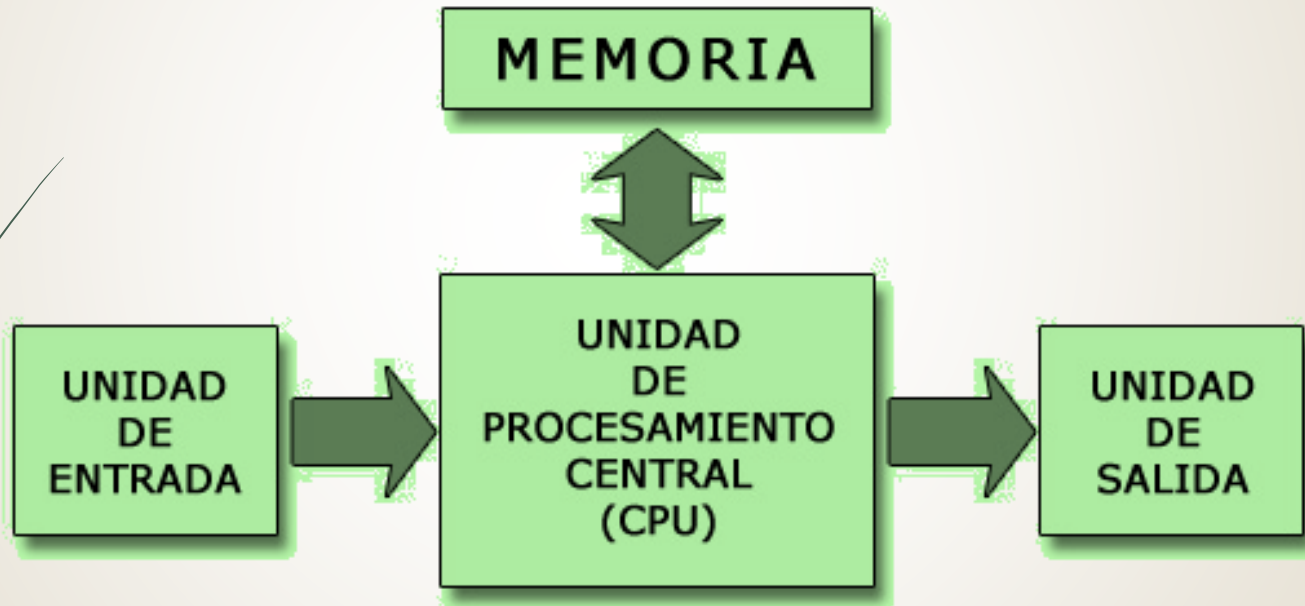


PROGRAMAS



Computadoras (2)

- Principales elementos de una computadora



Computadoras (3)

► Componentes

- Hardware (físico): procesador (CPU), memoria principal, memoria secundaria, dispositivos de entrada, salida y entrada/salida.
- Software (lógico): sistemas operativos, programas de aplicación.



Algoritmos

- Un algoritmo es una secuencia de pasos que de forma ordenada permite resolver un problema específico.
- Características
 - **Preciso** (indica el orden de pasos)
 - **Definido** (para la misma entrada siempre produce el mismo resultado)
 - **Finito** (tiene un N° finito de pasos)



Metodología

- Una metodología define los métodos, técnicas y herramientas que deben aplicarse al realizar un proceso.
- La *metodología de la programación* define cómo transformar un algoritmo en un programa de computadora.
- El *objetivo* de la programación es generar soluciones que automaticen tareas repetitivas para simplificar el trabajo rutinario de las personas.
 - Métodos: Indican la secuencia de tareas necesaria para realizar un trabajo.
 - Técnicas: Indican cómo realizar una tarea y las herramientas a utilizar.
 - Herramientas: Son los elementos que se aplican al realizar una tarea.



Solución por Computadora

➤ Fases

1. Análisis del Problema
2. Diseño del Algoritmo
3. Codificación
4. Compilación y Ejecución
5. Verificación y Depuración
6. Documentación y Mantenimiento



Análisis del Problema

- ➔ Se debe identificar claramente:
 - el problema a resolver,
 - las entradas de datos,
 - los resultados a obtener y
 - el objetivo del algoritmo.

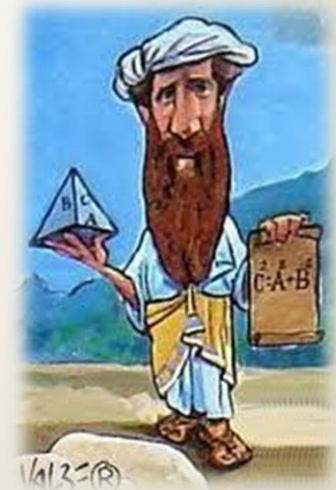


Problema
Calcular hipotenusa

Entradas
2 catetos

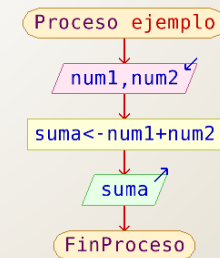
Salida
hipotenusa

Objetivo
Aplicar Pitágoras



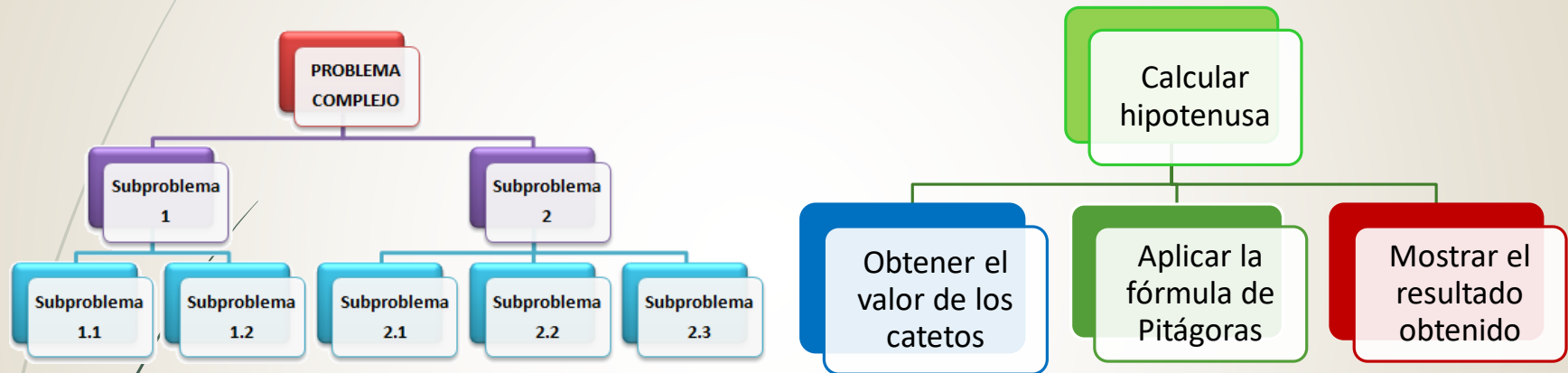
Diseño del Algoritmo (1)

- Se establecen los **pasos** y el **orden** en qué deben aplicarse para resolver el problema planteado.
- Descomposición del problema
 - Diseño Top-Down
 - Refinamiento Sucesivo
 - Herramientas de especificación de algoritmos (diagramas de flujo, diagramas N-S, pseudocódigo, fórmulas, lenguaje natural)

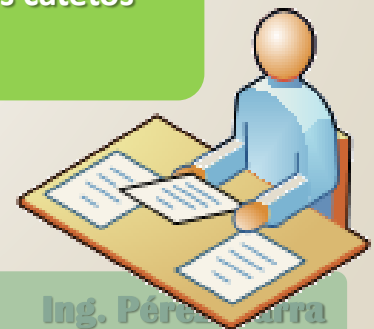
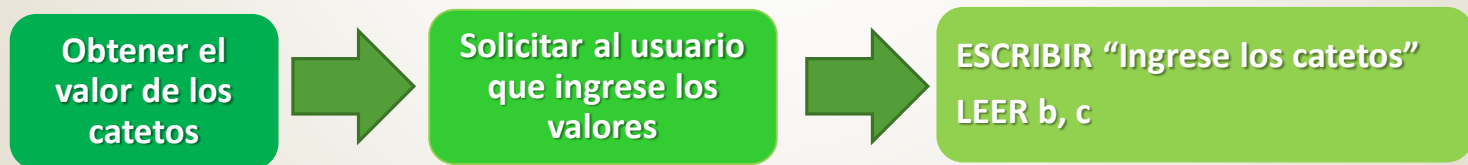


Diseño del Algoritmo (2)

Descomposición de Problemas



Refinamiento Sucesivo



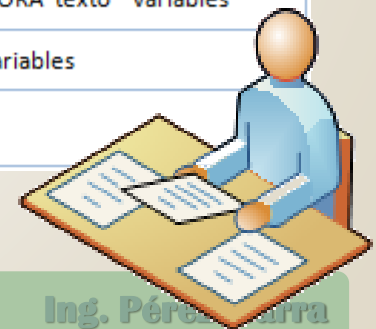
Diseño del Algoritmo (3)

Herramientas de Especificación de Algoritmos



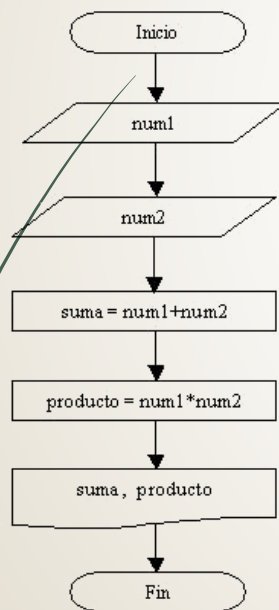
$$a = \sqrt{b^2 + c^2}$$

Diagrama de Flujo	Pseudocódigo
	INICIO ... FIN
	ESCRIBIR "texto" variables LEER variables
	variable ← operación
	SI condición ENTONCES ... FIN_SI
	variable ← función(par1, par2, par3) procedimiento(par1, par2, par3)
	ESCRIBIR PANTALLA "texto" variables
	ESCRIBIR IMPRESORA "texto" variables
	LEER TECLADO variables
	// comentarios



Codificación (1)

- Consiste en traducir el diseño del algoritmo a algún lenguaje de programación específico.



Lenguaje de Programación

```
1  \.LC0:
2      .string "Ingrese valor: "
3  \.LC1:
4      .string "Suma: "
5  \.LC2:
6      .string "Producto: "
7  main:
8      push    rbp
9      mov     rbp, rsp
10     sub     rsp, 16
11
12     main.cpp
13
14     1 // Online C++ compiler to run C++ program online
15     2 #include <iostream>
16     3 using namespace std;
17     4
18     5 int main()
19     6 {
20     7     int num1,num2,suma,producto;
21     8     cout << "Ingrese valor: ";
22     9     cin >> num1;
23    10     cout << "Ingrese valor: ";
24    11     cin >> num2;
25    12     suma=num1+num2;
26    13     producto=num1*num2;
27    14     cout << "Suma: ";
28    15     cout << "Producto: ";
29    16     return 0;
30 }
```

```
1  # Online Python compiler (interpreter) to run Python online.
2  # Write Python 3 code in this online editor and run it.
3  # coding: utf-8
4  # Your code here!
5  num1=int(input("Ingrese valor:"));
6  num2=int(input("Ingrese valor:"));
7  suma=num1+num2;
8  producto=num1*num2;
9  print("Suma: ",suma);
10 print("Producto: ",producto);
```

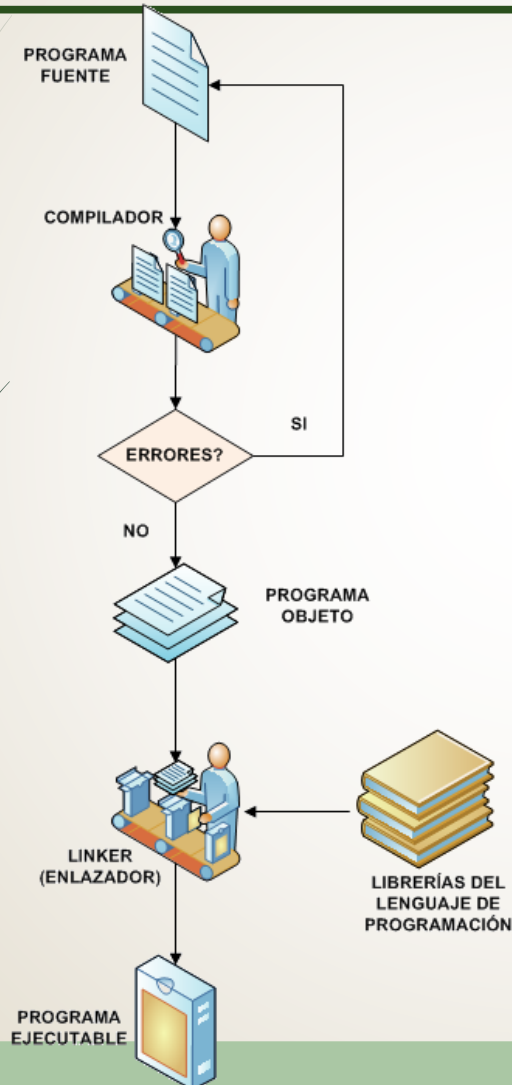
Codificación (2)

➤ Lenguajes de Programación

- Lenguaje Máquina: secuencias binarias (0's y 1's) que especifican instrucciones y datos directamente comprensibles por la computadora.
- Lenguajes de Bajo Nivel (ensamblador): instrucciones en especificadas códigos especiales llamados nemotécnicos (Por ejemplo, STAA \$0400)
- Lenguajes de Alto Nivel: diseñados para la fácil comprensión del ser humano.



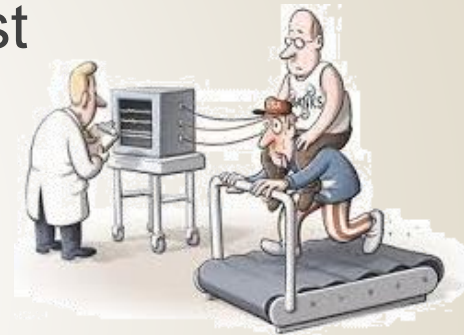
Compilación y Ejecución



- Un compilador es programa especial que traduce un programa fuente a lenguaje máquina (programa objeto).
- Un Linker (enlazador) es un programa que genera un programa ejecutable al agregar librerías (funciones especiales) a un programa objeto.

Verificación y Depuración

- La verificación comprueba el algoritmo a través de un conjunto de datos de prueba o test (valores normales, extremos y aspectos especiales).



- La depuración detecta, corrige y/o elimina errores en el programa.

- Errores de compilación
- Errores de ejecución
- Errores de lógica



Documentación y Mantenimiento

- Documentación Interna
 - Comentarios
- Documentación Externa
 - Manuales, diseños, análisis del problema
- Mantenimiento
 - Correctivo (localiza y **elimina errores** detectados luego de la entrega del programa)
 - Preventivo (**mejora** las **propiedades** de un programa sin alterar su funcionalidad)
 - Adaptativo (**adapta** un programa a nuevas condiciones del **entorno**)
 - Perfectivo (mejora o añade **nuevas funcionalidades** a un programa)

Conceptos Generales

- Partes de un Programa
 - Datos e Instrucciones
- Corrección de algoritmos
 - Cumplimiento con los requerimientos del problema
- Eficiencia de algoritmos
 - Calidad en el uso de recursos (tiempo, memoria)
- Reusabilidad de código



Bibliografía

- Sznajdleder, Pablo Augusto. Algoritmos a fondo. Alfaomega. 2012.
- López Román, Leonardo. Programación estructurada y orientada a objetos. Alfaomega. 2011.
- De Giusti *et al.* Algoritmos, datos y programas, conceptos básicos. Editorial Exacta. 1998.
- Joyanes Aguilar, Luis. Fundamentos de Programación. Mc Graw Hill. 1996.
- Joyanes Aguilar, Luis. Programación en Turbo Pascal. Mc Graw Hill. 1990.