



UNIVERSIDAD  
DE LOS ANDES  
VENEZUELA

***Programación Digital I***

---

---

---

# **Programación Digital I**

Gilberto Diaz  
gilberto@ula.ve  
Universidad de Los Andes  
Facultad de Ingeniería  
Escuela de Sistemas  
Depto de Computación  
Mérida - Venezuela



- Componentes Lógicos (Software)
  - S.O
  - Traductores
  - Aplicaciones
- ¿Qué es Programar?
- Lenguaje de Programación
- Niveles de programación.
- Paradigmas de Programación
- Metodologías de Programación
- Diseño de Algoritmos



Es la traducción de un algoritmo a un programa escrito en lenguaje de programación de alto nivel (programa fuente o código fuente)

Los diferentes pasos de un algoritmo se expresan en:

- instrucciones: término utilizado en lenguajes de bajo nivel
- sentencias o proposiciones: palabras utilizadas en lenguajes de alto nivel



En un programa encontramos los siguientes tipos de sentencias básicas:

- Sentencias de Entrada/Salida
- Sentencias Aritmético-lógicas
- Sentencias de decisión o selectivas
- Sentencias repetitivas o lazos



**Sentencias de Entrada/Salida:** Se utilizan para transferir información entre los diferentes dispositivos de entrada o salida (monitor, teclado , ratón, etc.) y la memoria principal

**Sentencias aritmético – lógicas:** se utilizan para efectuar operaciones aritméticas (suma, resta, multiplicación, etc) o lógicas (AND, OR)



**Sentencias de Decisión:** Permiten la selección de tareas alternativas sobre la base de expresiones condicionales

**Sentencias Repetitivas:** Se utilizan para realizar un conjunto de sentencias un número determinado o no determinado de veces.



## **Ejemplo:**

Inicio

Escribir “Introduzca el Ancho del paralelepípedo”

Leer ANCHO ...

```
float area {
```

```
float ancho, alto, largo, area;
```

```
printf(“Introduzca el Ancho del paralelepípedo\n”);
```

```
scanf(“%f”, &ancho);
```



## **Corrida en frío:**

Es la ejecución manual de cada sentencia del programa, utilizando un conjunto de datos de entrada determinados y verificando que los resultados son correctos.

Como una técnica de depuración, el programador debe utilizar datos que permitan recorrer todas las posibles rutas del programa.



Para el algoritmo de cálculo del área de un paralelepípedo:

ancho = 3.0

largo = 2.0

alto = 1.0

area =  $2*(3.0*2.0 + 3.0*1.0 + 2.0*1.0) = 22.0$



- El programa es escrito utilizando un editor de texto.
- Una vez completado, el programa es traducido por un compilador y el código objeto es generado si no hay errores de sintaxis.



- Al código objeto se le añaden funciones de librerías (otros programas) requeridas utilizando un programa llamado encadenador
- El código objeto, enlazado y cargado es ejecutado con los datos de entrada



- Es la acción de comprobar que el programa realice las tareas para las cuales ha sido diseñado y produce el resultado correcto y esperado.
- Si el programa tiene errores de lógica se deben corregir en el código fuente, compilar, ejecutar y revisar nuevamente.



- Fue diseñado a principio de los años 70 por Brian Kernigan y Dennis Ritchie.
- Combina características de los lenguajes de alto nivel (sentencias de manejo de datos y control) con características de los lenguajes de bajo nivel (manipulación de bits)
- Es independiente del hardware



- Es una extensión de C diseñado por Bjarne Stroustrup al principio de los años 80
- Es un lenguaje de programación híbrido donde se combinan las técnicas de programación estructurada y la programación orientada a objetos



- Es un enfoque que permite escribir programas estructurados.
- Se base en las siguientes estructuras de control:
  - Secuenciales (asignación, escritura y lectura)
  - Desición (simple, doble, múltiple)
  - Repetición (repita mientras, repita hasta, repita para)
- La unidad básica de este enfoque es “la función”



- Es un enfoque que encapsula datos (atributos) y métodos (comportamiento) en objetos.
- **Objeto:** Son componentes de software reutilizables que modelan objetos del mundo real
- Por lo general un objeto es un ente que puede ser percibido por alguno de los sentidos



- **Clases:** son declaraciones o abstracciones de objetos, lo que significa, que una clase es la definición de un objeto.
- Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase.
- Ejemplo:

Clase

Planeta

Objeto

júpiter, neptuno, marte



- En este curso se estudiará el enfoque estructurado porque los objetos se componen de fragmentos de programación estructurada
- Los métodos se implementan utilizando la programación estructurada



- Simple
- Sintaxis sencilla
- Flujo de control estructurado
- Tipos de datos variados
- Posibilidad de abstracción de datos
- Amplia variedad de operadores



- Palabras clave
- Identificadores
- Delimitadores
- Comentarios



for	while	do
float	int	char
double	signed	enum
extern	if	else
void	typedef	const
struct	sizeof	unsigned,
etc, etc		



## **consideraciones semánticas**

Son nombres que permiten referenciar o hacer mención de los diferentes componentes manipulados por un programa.

Deben ilustrar lo que representan, ejemplo

Si se está haciendo referencia al color de un gato el identificador podría ser: ColorGato



## **consideraciones sintácticas**

No pueden ser palabras reservadas

El primer caracter debe ser una letra

Se establece una diferencia entre letras mayúsculas y minúsculas

En la mayoría de los compiladores la máxima longitud es de 31 caracteres



## **Ejemplos**

Válidos:

Tempelnic, Contador, Pres1, aux, valor\_inicial

No Válidos

01val, -promedio, parámetro\_inic, ?aux



Son símbolos que permiten al compilador separar y reconocer los diferentes componentes del lenguaje

punto y coma (;) sirve para indicar la finalización de una sentencia

coma (,) sirve para separar los elementos consecutivos de una lista



paréntesis () enmarcan una lista de parámetros, expresiones o condiciones

corchetes [] enmarcan las dimensiones de arreglos

llaves {} enmarcan un conjunto de sentencias de una estructura dada



Sirven para documentar el programa o agregar notas importantes que describen las funciones de un conjunto de sentencias

Estos son delimitados por las secuencias de caracteres `/* */`

Estos no agregan peso extra al programa.



Sirven para documentar el programa o agregar notas importantes que describen las funciones de un conjunto de sentencias. Enriquecen el código

Estos son delimitados por las secuencias de caracteres `/* */`. Ejemplo:

```
/* Funcion para el calculo del area */
```



Estos no agregan peso extra al programa.

Deben ser coherentes con el programa,  
relevantes

Deben mantenerse actualizados junto con el  
código



El profesor ahora mostrará un ejemplo de código e ilustrará su compilación y ejecución