

## UNIDAD 1: INTRODUCCIÓN GENERAL

### \* PROGRAMA

Un **programa** es un conjunto de instrucciones para realizar una actividad o resolver un problema, se deben ejecutar una serie de acciones sobre determinados datos.

Un programa se compone de : Estructuras de datos + Operaciones primitivas elementales + Estructuras de control

### \* LENGUAJE DE PROGRAMACIÓN

Es el medio a través del cual le comunicamos a la computadora **la secuencia de instrucciones que debe ejecutar para llevar a cabo actividades.**

Todo lenguaje debe manejar los elementos de un programa: estructura de datos, operaciones primitivas elementales y estructuras de control.

### CARACTERÍSTICAS

Todo lenguaje está compuesto por un alfabeto, un vocabulario y una gramática o sintaxis.

**Alfabeto:** conjunto de caracteres (alfabéticos, numéricos, especiales)

**Vocabulario:** conjunto de palabras válidas o reservadas. Ej. En Java: if, then, else, do, for, while, entre otras.

**Gramática o sintaxis:** conjunto de reglas para construir las instrucciones.

### \* PROCESO DE PROGRAMACIÓN

Elaborar un programa implica llevar a cabo una **serie de pasos secuenciales y cronológicos** que comienzan con la detección y definición del problema y conducen a la implantación del programa que lo soluciona.

### PASO 1: DEFINICIÓN DEL PROBLEMA

Identificar el problema, establecer los límites.

Ej: Situación: Calcular el sueldo de un empleado.

La solución debe ser el monto a pagar.

Cada empleado tiene un sueldo asignado por unidad de tiempo trabajada y un tiempo trabajado.

### PASO 2: ANÁLISIS DEL PROBLEMA

Entender en detalle el problema en cuestión en términos de **datos de entrada disponibles, proceso necesario y la información resultante.**

Información esperada:

- > Nombre Empleado: Juan García
- > Sueldo: \$ 1.800 mensuales

Datos a identificar:

Nombre empleado, Cantidad de días trabajados, Valor sueldo por día trabajado

Determinar el proceso:

¿Cómo se determina el nombre del empleado? No se calcula, sólo se ingresa.

¿Cómo se calcula el sueldo mensual? Es un dato que se obtiene multiplicando los días trabajados por el valor diario.

### PASO 3: DISEÑO DEL PROBLEMA

Se diseña la lógica para solucionar el problema, para lo cual debemos hacer:

**El algoritmo: secuencia ordenada y cronológica de pasos que llevan a la solución del problema.**

Deben ser: simples, precisos, ordenados, tener principio y fin.

La prueba: se simula el funcionamiento del algoritmo con datos propios para validar la correcta operación del algoritmo.

### PASO 4: CODIFICACIÓN DEL PROBLEMA

Se codifica el programa con el lenguaje de programación elegido. Sólo debemos convertir las acciones del algoritmo en instrucciones del lenguaje.

El programa codificado debe editarse, compilarse, probarse y depurarse.

Para que un programa lo entienda y ejecute una computadora debe estar en lenguaje de máquina o código objeto, es decir el programa debe ser traducido por un compilador (traductor del lenguaje).

## PASO 5: **IMPLANTACIÓN DEL PROBLEMA**

Una vez correcto el programa, se instala y se pone a funcionar, entrando en operación o producción.

## PASO 6: **MANTENIMIENTO DEL PROBLEMA**

El programa instalado debe ser supervisado continuamente para detectar posibles cambios o ajustes necesarios. Por otro lado puede requerir adecuaciones de datos o proceso determinados por la dinámica del problema o la organización.

### \* EL ALGORITMO

Podemos aplicar el concepto de algoritmo a situaciones de nuestra vida cotidiana, tales como seguir una receta de cocina, instrucciones para un juego, etc.

Siempre vamos a seguir los 3 primeros pasos del proceso:

- > 1- Definir el problema
- > 2- Analizar el problema
- > 3- Diseñar el programa

### EJ.: CAMBIAR FOCO QUEMADO

1. Colocar escalera
2. Subir escalera
3. Quitar foco quemado
4. Bajar escalera
5. Obtener foco nuevo
6. Subir escalera
7. Colocar foco nuevo
8. Bajar de la escalera
9. Guardar la escalera
10. Fin

### \* TÉCNICAS DE DISEÑOS DE PROGRAMAS

**Diagrama de Flujos u organigrama:** técnica gráfica que sirve para expresar el orden en que deben ejecutarse las instrucciones

**Pseudocódigo:** técnica que emplea palabras claves en castellano (sentencias) que indican la operación a realizar.

### \* ESTRUCTURA GENERAL DE UN PROGRAMA

Debemos tratar que la escritura de un programa, mediante una **herramienta de programación**, sea lo más **clara** posible **y estructurada**, de modo que su lectura facilite considerablemente el entendimiento del programa, cuando se realice su validación, en el momento de su codificación en un lenguaje de programación y en su posterior mantenimiento.

**Programa** nombredelprograma

**VAR** declaración de variables

**INICIO**

Acciones

**FINPROGRAMA**

Todas las palabras en negrita se denominan **palabras reservadas**.

Donde:

Nombredelprograma: sigue las mismas reglas de una variable.

Declaración de variables: se enumeran las variables que se usan en el programa, el formato es el siguiente:

VAR nomvar1, nomvar2,...: tipo de variable1

VAR nomvar3, nomvar4,...: tipo de variable2

### DECLARACIÓN DE VARIABLES

Ej:

**VAR superficie, volumen: ENTERO**

**nombre, apellido: CADENA**

**estatura: REAL**

**O bien:**

**VAR superficie, volumen: ENTERO; nombre, apellido: CADENA; estatura: REAL**

## COMENTARIOS

Para darle claridad y organización a los programas **para su posterior mantenimiento** es recomendable el uso de comentarios dentro del programa.

Se indica con // y puede estar en una línea independiente o ser parte de otra.

Ej:

**//calcular la cantidad de elementos**  
**contador=1 //inicializa el contador**

## OPERACIÓN DE ASIGNACIÓN

Es una acción para darle valor a una variable:

**Formato general: variable = expresión/variable/constante**

La acción de asignar es destructiva, es decir que el valor anterior que tuviera se pierde.

No se puede asignar valores a una variable de un tipo diferente del suyo.

## ENTRADA Y SALIDA DE INFORMACIÓN

Las operaciones de entrada permiten leer valores (desde periférico de entrada) y asignarlos a variables, se conocen como operaciones de lectura.

**Leer (lista variables entrada )**

Las operaciones de salida permiten mostrar valores (en periférico de salida), se conocen como operaciones de escritura.

**Escribir (lista variables/constantes/expresiones)**

## \* ESTILO DE PROGRAMACIÓN

Además de seguir reglas de sintaxis de un lenguaje y las técnicas básicas de programación, es cuestión de tener reglas de disciplina y orden en el diseño de programas.

Un buen estilo de programación hace a su legibilidad, a la calidad, mejor comprensión.

## CALIDAD DE UN PROGRAMA

Parámetros a considerar:

Un programa debe funcionar correctamente.

La documentación es importante para ayudar a comprender y utilizar un programa.

La eficiencia viene dada por la relación tiempo-espacio.

La corrección: un programa debe proporcionar los resultados correctos.

La flexibilidad: los programas deben permitir cambios con ligeros toques.

La fiabilidad: la dan la exactitud y precisión de los resultados.

La presentación: para hacer los programas legibles es importante usar comentarios y sangrado.

## **UNIDAD 2: TIPOS DE DATOS, OPERACIONES Y EXPRESIONES**

### \* ESTRUCTURAS DE DATOS

Dijimos que :

Programa= Estructura de datos + Operaciones primitivas elementales +  
Estructuras de control

Un dato es la representación de los objetos con los que opera una computadora.

Las estructuras de datos: representación interna de datos en la computadora.

Pueden manejarse como variables o como constantes.

### \* TIPOS DE DATOS SIMPLES

Numérico

> Entero

> Real

Cadena de caracteres

> Alfabéticos

> Alfanuméricos

Carácter

Lógico

## DATOS NUMÉRICOS

**Enteros:** son los que no contienen componentes decimales, es decir no incluyen el punto decimal, pueden ser positivos o negativos.

**Ej: 450; -325; 4; - 4**

**Reales:** son los que contienen una parte decimal, es decir incluyen el punto decimal. Pueden ser positivos o negativos.

**Ej: 465.25; 800.02; -24.5; - 1036.44**

## DATOS CADENA DE CARACTERES

Compuesto por una **serie de caracteres alfabéticos, numéricos y especiales.**

Sirven para representar nombres de personas, descripciones, direcciones.

Los valores son alfabéticos o alfanuméricos y se delimitan por comillas Ej:

**“Universidad Tecnológica Nacional”, “Rodríguez 273 – Ciudad”**

## DATOS CARÁCTER

Es **un solo carácter**, utiliza una sola posición. Puede ser cualquier carácter del código ASCII.

**Ej: “A”, “#”, “I”**

## DATOS LÓGICOS

Este tipo de dato está compuesto por los **valores lógicos: Verdadero o Falso.**

Se utiliza generalmente cuando se debe manejar estructuras lógicas de control.

Se determinan como: **[V] o [F]**

## \* VARIABLES

Son valores que **cambian a lo largo del programa.**

Todo dato que se ingresa o se genera o se calcula a partir de otros datos, debe identificarse y manejarse en forma de variable.

Una variable tiene las siguientes características:

**Nombre:** identificador de la variable. Sirve para referenciarla.

**Contenido:** valor de la variable.

**Tipo de dato:** especifica qué tipo de dato es.

## NOMBRES DE VARIABLES: REGLAS

-Se puede usar combinaciones de letras mayúsculas y minúsculas, dígitos y guión bajo.

-El nombre debe iniciar con una letra.

-No debe ser palabra reservada. Ej: SI, ENTONCES, SINO, PARA, HACER, ...

-Es conveniente que sea significativa y con relación al objeto que representa.

-Tener en cuenta la longitud máxima de caracteres, por ej. 10.

### **Nombres válidos:**

**Nombre\_Empl, Nombre1, J**

### **Nombres no válidos o no adecuados:**

**1Nombre, nnnnnnnnn, 123, Pesos\$**

## \* CONSTANTES

Son valores específicos, es decir que **no varían durante la ejecución** del algoritmo.

Ejemplo: **34, “Mes”, “52”**

## \* OPERACIONES PRIMARIAS ELEMENTALES

Son las **acciones básicas** que la computadora puede ejecutar.

- > Declaraciones
- > Lectura de datos (entrada)
- > Operaciones aritméticas fundamentales
- > Escritura de datos (salida)

## DECLARACIONES

Es una **acción no ejecutable.**

Se declaran todas las variables, constantes y tipos de datos que se utilizarán en el algoritmo.

## DECLARACIONES DE CONSTANTES

Formato:

***CONST***

***NOMCONSTANTE = Valor***

Donde:

***NOMCONSTANTE*** : Identificador de la constante

***Valor*** : Valor de la constante

Ej:

**CONST**

**PI = 3.145926**

**MAXIMO = 100**

**RESULTADO1 = “Aprobado”**

**RESULTADO2 = “Desaprobado”**

## DECLARACIONES DE VARIABLES

Formato:

***VAR***

***nomVariable1 : TIPO DE DATO***

***nomVariable2 : TIPO DE DATO***

Donde:

***nomVariable1, nomVariable2*** : Identificador de la variable

***Tipo de dato*** : tipo de dato que tendrá la variable

Ej.:

**VAR**

**nombreEmpl: CADENA**

**valorHora: REAL**

## LECTURA DE DATOS (ENTRADAS)

Permite **introducir los datos** para realizar el proceso, necesarios para el algoritmo.

La entrada de datos puede hacerse desde cualquier dispositivo como teclado, lector de código de barras, mouse, o cualquier otro.

Formato:

***Leer (nomVar1, nomVar2, ...)***

Donde:

**Leer** : *indica la acción*

**nomVar1, nomVar2,...** : *nombres de las variables donde se leerán los datos*

Si tuviéramos que solicitar el nombre del empleado, sus horas trabajadas y el valor por hora; seria:

**Leer (nombreEmpleado, horasTrab, valorHora)**

También puede ser:

//Solicitar Nombre empleado

**Leer (nombreEmpl)**

//Solicitar Horas trabajadas

**Leer (horasTrab)**

//Solicitar Valor hora

**Leer (valorHora)**

## OPERACIONES ARITMÉTICAS FUNDAMENTALES

Estas operaciones **transforman los datos**, los procesan para convertirlos **en información**. Con las operaciones se forman expresiones aritméticas para realizar cálculos.

Formato:

***variable = expresión***

Donde:

**variable** : nombre de la variable donde se asignará el resultado de la expresión, entero o real.

**expresión** : es una constante, variable o expresión algebraica; cuyo resultado se asigna a la variable.

**=** : Es el símbolo que indica asignación del valor de la expresión de la derecha a la variable de la izquierda.

EXPRESIONES

Las expresiones son **combinaciones válidas de operandos y operadores**, es decir combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

Cuando en una expresión las variables toman valor, puedo obtener el resultado de la expresión.

Las variables y constantes dentro de una expresión se denominan operandos.

TIPOS DE EXPRESIONES

- >Aritméticas
- >Lógicas
- >De relación
- >De carácter

OPERADORES

**Operadores aritméticos:**

- Suma: +
- Resta: -
- Multiplicación: \*
- División: /
- Exponenciación: ^

**Operador de concatenación:**

- + (Se distingue de la suma por los operandos)

PRIORIDADES

- 1. ^
- 2. \*, /
- 3. +, -
- 4. <, >, <=, >=, =
- 5. [NO]
- 6. [Y]
- 7. [O]

**La presencia de paréntesis, cuya prioridad es 0, indica que se debe empezar resolver por los paréntesis de más adentro y luego ir resolviendo hacia afuera.**

EXPRESIONES ARITMÉTICAS

Las expresiones aritméticas deben escribirse en una sola línea para que lo reconozca la computadora, y evalúe cada componente en un cierto orden de precedencia (prioridad) que tienen las operaciones aritméticas.

Ej: **n = (x + y) / (y - 2)**

**variable = expresión aritmética**

Para calcular el Sueldo que equivale a las horas trabajadas por el valor de la hora.

Se escribiría:

**sueldo = horasTrab \* valorHora**

Otros ej.:

- A = 1** // A toma el valor 1
- B = A + 1** // B toma el resultado 2

OPERADORES LÓGICOS

Permiten formular **condiciones complejas** a partir de condiciones simples, son:

- Conjunción [AND]: [Y]
- Disyunción [OR]: [O]
- Negación [NO]: [NO]

## TABLA DE OPERADORES LÓGICOS

P	Q	[NO] P	[NO] Q	P [O] Q	P [Y] Q
V	V	F	F	V	V
V	F	F	V	V	F
F	V	V	F	V	F
F	F	V	V	F	F

## EJEMPLO DE EXPRESIONES

sup, base, perimetro: ENTERO

sup= 10, base=2, perímetro = 14

Expresión	Resultado
<b>sup*base*sup+10 &gt; perimetro [0] perimetro &gt; 10</b>	1

Orden resolución:      **2      2      3      4                      7                      4**

Resolución:

$$\begin{array}{rcll} 10 & * & 2 & \\ 20 & & * & 2 \\ 40 & & + & 10 \\ 50 & & & > 14 \\ & & & [V] \end{array}$$

[O]  $14 > 10$   
[V]

Resultado: [V]

### ESCRITURA DE DATOS (SALIDA)

Mediante la escritura damos **salida a los datos** de la computadora hacia un medio periférico, como por puede ser la pantalla, la impresora, entre otros.

Formato:

***Escribir (nomVar1, nomVar2, ... nomVarN)***

Donde:

**Escribir:** indica la acción de mostrar

**nomVar1, nomVar2,...:** nombres de las variables que contienen los datos que serán mostrados

Ej.:

**Escribir (nombreEmpl, sueldo)**

O bien si queremos un mensaje:

**Escribir(“Nombre:”, nombreEmpl, “Sueldo: \$”, sueldo)**