

Universidad Centroccidental

“Lisandro Alvarado”

Decanato de Ciencias y Tecnología



Unidad I: Introducción a la Programación

Tema 3: Programación

Coordinación de Introducción a la Computación
Prof. Gisela Parra

Contenido:

1. El Software.

- **Definición**
- **Clasificación según su función:**
 - ✓ Sistemas Operativos.
 - ✓ Programas de Aplicación.
 - ✓ Lenguajes de Programación.
- **Tipos de Lenguajes de Programación:**
 - ✓ De Máquina.
 - ✓ De Bajo Nivel – Ensamblador.
 - ✓ De Alto Nivel –
Intérpretes/Compilador.
- **Pasos para la Ejecución de un Programa.**
- **Interacción Programa-Computador**
- **Interacción Humano – Computador**
- **Instrucciones de Programas y el CPU**
- **Los Datos en la memoria RAM**

Contenido:

2. Algoritmos.

- **Definición y Características.**
- **Tipos:**
 - ✓ Lenguaje Natural.
 - ✓ Seudolenguaje.
 - ✓ Lenguaje Formal.
- **Ejemplos de Algoritmos en Lenguaje Natural como solución a Problemas Cotidianos.**

3. Programación.

- **Definición.**
- **Símbolos básicos de carácter múltiple**
- **Tipos de Datos.**
- **Elementos básicos**
 - ✓ Identificadores
 - ✓ Constantes
 - ✓ Variables
- **Operadores y su Orden de Evaluación**
- **Expresiones**
 - ✓ Aritméticas
 - ✓ Lógicas
- **Sistemas Numéricos**
- **Proposiciones**
 - ✓ Asignación y almacenamiento
 - ✓ De Bifurcación y Toma de decisiones
 - ✓ Programación Iterativa

3. Programación

Definición:



- Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación.
- El propósito de la programación es crear programas que exhiban un comportamiento deseado.
- El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal.

3. Programación

Símbolos básicos de Carácter Múltiple

Operaciones Básicas

Lectura /Escritura

Entrada:

leer identificador

Ejemplo:

leer nota

Salida:

escribir “Mensaje”

escribir Identificador

Ejemplo:

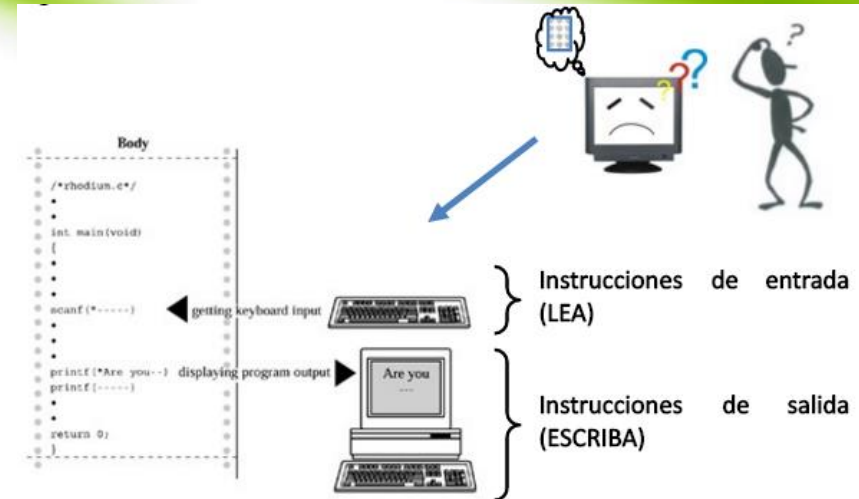
escribir “La nota es:”

escribir nota

ó

escribir “La nota es: ”, nota

Unidad I :Introducción a la Computación.



3. Programación

Operaciones Básicas Asignación

Formato:

Identificador2 = Identificador1

Ejemplos:

`nombreAlumno = nombre` Se asigna el contenido de la variable `nombre` a la variable `nombreAlumno`

Identificador = valor

Ejemplo:

`nota = 20` Se asigna el valor 20 a la variable `Nota`
`area = lado*lado`

3. Programación

Tipos de Datos:

entero: Números sin punto decimal.

Ejemplos: 234 567 100

real: Contienen punto decimal.

Ejemplos: 234.78 567.56 100.12

alfanumérico: Cualquier valor contenido en la tabla ASCII ó combinación de letras y números.

Ejemplos: José, @gmail.com, #EstudianteUCLA, V-12345678

lógico: verdadero, falso

Ejemplo: True, False

3. Programación

Elementos Básicos:

- Dato e Información
- Identificador
- Dato Variable
- Dato Constante

Dato e Información :

Dato	Información
<ul style="list-style-type: none">• Representación Simbólica• No tienen sentido semántico• No transmiten Mensaje• Describen situaciones, hechos <p>Ejemplo: 18</p>	<ul style="list-style-type: none">• Conjunto de Datos procesados• Conjunto de Datos organizados• Tienen un significado• Transmiten un mensaje• Permite la toma de decisiones• Favorece a la resolución de problemas• Incrementa el conocimiento <p>Ejemplo: 18 es la nota obtenida en el primer parcial de Cálculo.</p>

3. Programación

Identificador

- Palabra definida por el usuario para denotar cualquier dato o elemento de un algoritmo, se utiliza para referenciar cualquier dirección de memoria donde se van a almacenar los datos.
- El identificador de una variable debe contener la primera letra en minúscula y si es compuesto por dos palabras la inicial de la segunda palabra se indica en mayúscula, este **tipo de notación es denominado dromedaryCase.** En caso de constantes se indica todo el nombre en mayúscula.

3. Programación

Reglas para los Identificadores

- Debe comenzar por una letra en minúscula, tal como lo expresa la notación dromedaryCase.
- Puede ser combinación de letras y dígitos
- No debe llevar espacios en blanco
- No se debe utilizar los operadores aritméticos $*$, $-$, $+$, $/$, $%$, ni palabras reservadas.
- No deben tener el mismo nombre que otras variables cuyas declaraciones aparezcan en el mismo ámbito.
- Usar nombres legibles y referentes a lo que se va a utilizar

3. Programación

Dato Variable

El contenido cambia, se modifica o transforma medida que se ejecuta un programa.

Ejemplos:

entero	numeroHijos
real	sueldo, promedio
lógico	encontrado
alfanum	cedulaEmpleado

3. Programación

Dato Constante

El valor no se modifica durante toda la ejecución del programa.

Formato:

const tipo NOMBRE = valor

Ejemplos:

```
const real PI = 3.1416
```

```
const entero MESES = 12
```

3. Programación

Operadores y su Orden de Evaluación

Expresiones Aritméticas: Son expresiones o fórmulas que se plantean para lograr encontrar la solución de un problema matemático.

Operadores y Significado:

OPERADORES ARITMETICOS			
Operador	Descripción	Ejemplo	Resultado
+	Suma	$c = 3 + 5$	$c = 8$
-	Resta	$c = 4 - 2$	$c = 2$
-	Negación	$c = -7$	$c = -7$
*	Multiplicación	$c = 3 * 6$	$c = 18$
**	Potenciación	$c = 2 ** 3$	$c = 8$
/	División	$c = 7.5 / 2$	$c = 3.75$
//	División entera	$c = 7.5 // 2$	$c = 3.0$
%	Módulo	$c = 8 \% 3$	$c = 2$

Las expresiones aritméticas son evaluadas de acuerdo a la prioridad de sus operadores:

Prioridad	Orden de evaluación
1. ()	
2. * / , %	Lo primero que aparezca de izquierda a derecha
3. + , -	Lo primero que aparezca de izquierda a derecha

3. Programación

Determinar el valor de las siguientes expresiones aritméticas:

b) $6+2*(9+3)/3-3\%2$

Respuesta:

$$6+2*(9+3)/3-3\%2$$

$$6+2*12/3-3\%2$$

$$6+24/3-3\%2$$

$$6+8-3\%2$$

$$6+8-1$$

$$14-1$$

$$13$$

3. Programación

Actividad Propuesta

Determinar el valor de las siguientes expresiones aritméticas:

- a) $(4+40/5)\%3$
- b) $13-(26+2*5)/4\%3$
- c) $3*10+4*(8+4*7-10*3)/6$
- d) $10\%5$
- e) $6+2*(9+3)/3-3\%2$
- f) $(4+40/5)\%3$
- g) $13-(26+2*5)/4\%3$

3. Programación

Linealidad en las Expresiones Aritméticas

Toda expresión algebraica debe ser representada en forma lineal para ser procesada en un lenguaje de programación

Ejemplo:

$$\frac{3x + y}{z} = (3 * x + y) / z$$

3. Programación

Expresiones Lógicas

Las **expresiones lógicas** son operaciones que dan como resultado dos posibles valores falso (F) o verdadero (V). Los **operadores lógicos básicos** son NOT (no) , AND (y) y OR (o)

NOT (\sim)		AND (\wedge)			OR		
a	$\sim a$	a	b	a AND b	a	b	a OR b
F	V	F	F	F	F	F	F
V	F	F	V	F	F	V	V
		V	F	F	V	F	V
		V	V	V	V	V	V

$(A - 1) \text{ AND } (M \leq 1) \text{ OR } (C == 0)$

$(A > B) \text{ OR } (F \neq (C + 3))$

3. Programación

Prioridad de los Operadores Lógicos y Relacionales

Al igual que los operadores aritméticos, los operadores lógicos y relacionales presentan un orden de prioridad, la siguiente tablea resume dicho orden

Prioridad	Operación
1	<, >, <=, >=, ==, !=
2	NOT
3	AND
4	OR

$(A \neq 1) \text{ AND } (M \leq 1) \text{ AND } ((M \geq 0) \text{ OR } \text{NOT}(C == 1))$
 $((A > B) \text{ OR } (F \neq 1)) \text{ AND } (M < N)$
 $(1 > 0) \text{ OR } (-3 \leq -5) \text{ AND } (-5 > -1) \text{ AND } (100 == 1000) \text{ OR } (1 \neq 0)$

3. Programación

Sistemas Numéricos: Un sistema numérico es un conjunto de reglas que permiten representar un número con una cantidad finita de símbolos. El número de símbolos que utiliza un sistema numérico se denomina base.

Sistema Numérico Decimal:

Base = 10

Compuesto de 10 dígitos (0 al 9)

Sistema Binario:

Base = 2


Compuesto por 2 dígitos(0 y 1)

3. Programación

Sistemas Numéricos

Conversión de Binario a Decimal

Se enumeran las posiciones del número binario de derecha a izquierda comenzando desde 0. El resultado se obtiene multiplicando el dígito binario por la base del sistema elevado a la posición en la que se encuentra.

Posiciones 

76543210
↑↑↑↑↑↑↑↑

Ejemplo: $(0100\ 0000)_2 = 1 \times 2^6$
 $= (64)_{10}$
 $(0100\ 0000)_2 = @$

3. Programación

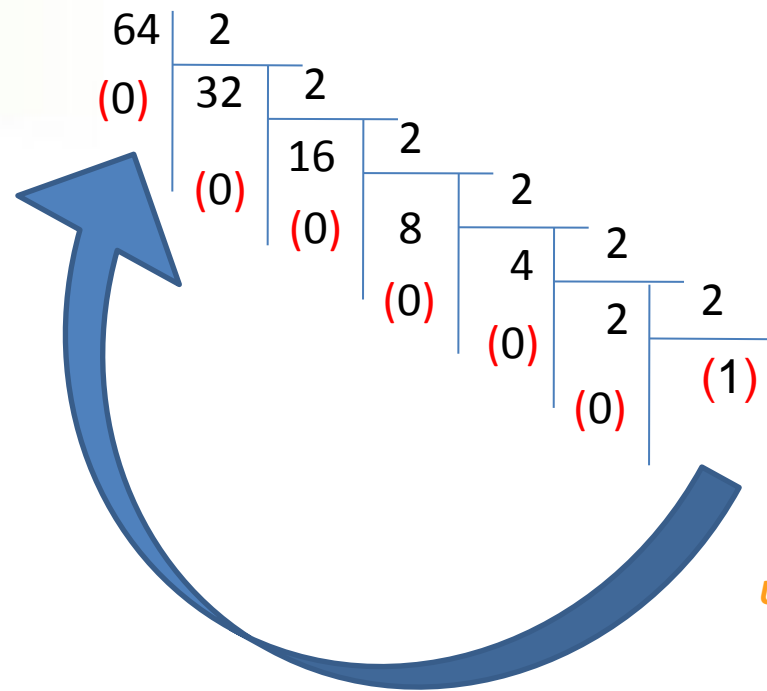
Sistema Numérico Binario:

Conversión de Decimal a Binario

Para obtener el equivalente se divide sucesivamente entre 2 hasta que el cociente sea 1. El número binario equivalente se forma tomando el último cociente y todos los residuos obtenidos, desde el último al primero.

Ejemplo

$$64_{10} = (X)_2$$



$$64_{10} = 100\ 0000_2$$

$$64_{10} = (0100\ 0000)_2$$

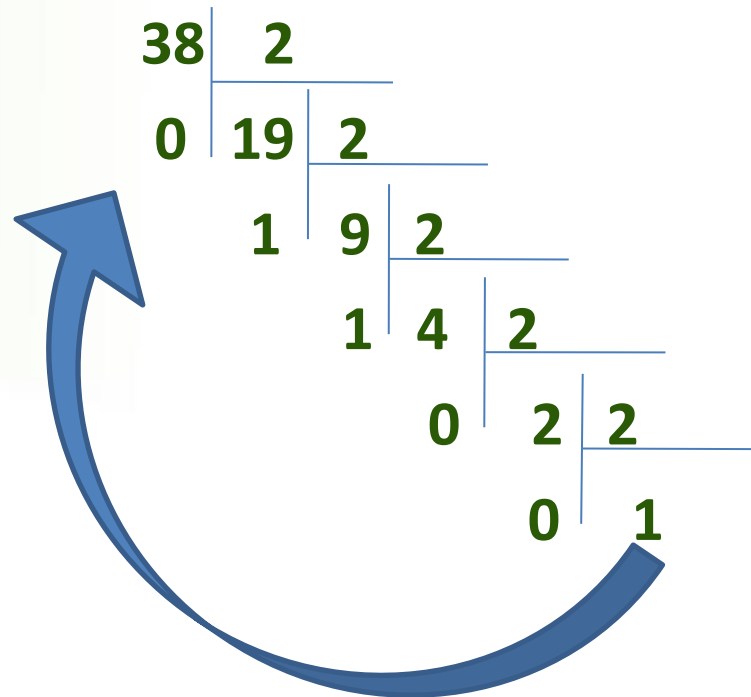
Unidad I :Introducción a la Computación.

3. Programación

Sistema Numérico Binario

Realizar las siguientes conversiones:

a. $(38)_{10} = (?)_2$



$$(38)_{10} = (0010\ 0110)_2$$

4. Programación

Sistema Numérico Binario

Realizar las siguientes conversiones:

b. $(00100101)_2 = (?)_{10}$

Posiciones ←

76543210

↑↑↑↑↑↑↑↑

$$\begin{aligned}(00100101)_2 &= 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^0 \\ &= 32 + 4 + 1\end{aligned}$$

$$(00100101)_2 = (37)_{10}$$

3. Programación

Sistema Numérico Binario

Realizar las siguientes conversiones:

b. $(00100101)_2 = (?)_{10}$

Posiciones ←

76543210

↑↑↑↑↑↑↑↑

$$\begin{aligned}(00100101)_2 &= 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^0 \\ &= 32 + 4 + 1\end{aligned}$$

$$(00100101)_2 = (37)_{10}$$

3. Programación

Ejercicios Propuestos

Convertir al sistema que se le indica:

- a. $(67)_{10} = (?)_2$
- b. $(112)_{10} = (?)_2$
- c. $(38)_{10} = (?)_2$
- d. $(00101100)_2 = (?)_{10}$
- e. $(01011111)_2 = (?)_{10}$
- f. $(00010000)_2 = (?)_{10}$

3. Programación

Proposiciones de Asignación y Almacenamiento

Operador de asignación (\leftarrow)

variable \leftarrow expresión aritmética, variable o una constante

Ejemplo:

Edad1 \leftarrow 15

Edad2 \leftarrow 30

Edad3 \leftarrow Edad2

Suma \leftarrow Edad1+Edad2+Edad3

(¿Qué valor se almacena en Suma?)

3. Programación

Proposiciones de Bifurcación y Toma de Decisiones

Selectiva Simple: Se utiliza cuando la opción que se va a efectuar depende de la evaluación de sólo una condición.

si Condición entonces

acción

sino

acción;

Ejemplo :

si $\text{Nota} \geq 48$ entonces

escribir “El estudiante aprobó”

sino

escribir “El estudiante reprobó”

3. Programación

Programación Iterativa

Se emplean las **estructuras iterativas** cuando necesitamos que un proceso se repita una cantidad de veces, podríamos repetir el proceso manualmente pero obviamente este algoritmo no sería el óptimo.

Estructuras :

MIENTRAS

Fin_mientras

PARA

Fin_para

3. Programación

Programación Iterativa

Mientras

Fin_mientras

Ejemplo

- El problema es calcular la suma de los números naturales desde 1 hasta n .
- **Inicio**
- Entero i, n, Suma
- Leer(n)
- $i=1$
- $\text{Suma} = 0$
- **MIENTRAS** ($i \leq n$)
- $\text{Suma} = \text{Suma} + i$
- $i = i + 1$
- **Fin_mientras**
- Escribir (Suma)
- **Fin**

3. Programación

Programación Iterativa

PARA

Fin_para

Ejemplo

- El problema es calcular la suma de los números naturales desde 1 hasta **n**.
- **Inicio**
- Entero i, n, Suma
- Leer(n)
- $i=1$
- Suma =0
- **PARA** $i=1$ hasta n hacer
- Suma=Suma+i
- **Fin_para**
- Escribir (Suma)
- **Fin**

Universidad Centroccidental

“Lisandro Alvarado”

Decanato de Ciencias y Tecnología



Unidad I: Introducción a la Programación

Tema 3: Programación

Coordinación de Introducción a la Computación
Prof. Gisela Parra