

UNIVERSIDAD TECNOLÓGICA PRIVADA DE SANTA CRUZ

FACULTAD DE CIENCIAS TECNOLÓGICAS

GUÍA



INTRODUCCIÓN A LA PROGRAMACION

Santa Cruz, 2010

PROLOGO

Esta asignatura está orientada al conocimiento de los fundamentos teóricos de la programación de las computadoras y al desarrollo del pensamiento lógico para descomponer problemas sencillos y de mediana complejidad en pasos ordenados a seguir para su resolución que es lo que se define como algoritmo.

Además, se describen las estructuras básicas de los algoritmos y se representan mediante los modelos de diagramas de flujo (a través de símbolos gráficos) y pseudocódigo (lenguaje coloquial) este último orientado a un lenguaje de programación.

JOSÉ de JESÚS CASTRO MEDINA

RESPONSABLE de DISCIPLINA de
ALGORITMOS y PROGRAMACIÓN

NOMBRE DEL DOCENTE

I. DATOS PERSONALES

Fecha de Nacimiento :
Profesión :
Nacionalidad :
Estado Civil :

II. EDUCACIÓN

Secundaria :
Universidad :
Estudios Superiores :

III. EXPERIENCIA LABORAL

.....
.....
.....
.....
.....
.....
.....

IV. EXPERIENCIA DOCENTE

.....
.....
.....
.....
.....

UNIVERSIDAD TÉCNICA PRIVADA DE SANTA CRUZ
Facultad de Ciencias Tecnológicas
Sistema Presencial Universitario

MATERIA	INTRODUCCIÓN A LA PROGRAMACIÓN	SIGLA	SIS-110
CARRERA	Ing. Informática y Sistemas Administrativos Ing. Electrónica y Sistemas Ing. Redes y Telecomunicaciones Ing. Sistemas	SEMESTRE	Segundo
		HORAS	80
		HORAS TEÓRICAS	40
PRERREQUISITO	ADMISIÓN	HORAS PRÁCTICAS	60

OBJETIVO GENERAL

- *CAPACIDAD DE ANALIZAR Y SINTETIZAR UN PROBLEMA EN PARTES CON EL FIN DE DISEÑAR SU REPRESENTACIÓN ALGORÍTMICA*

PLAN TEMÁTICO

Unidad	Temas	Tiempo Teoría (Minutos)	Tiempo Práctica (Minutos)
I	SISTEMAS DE NUMERACIÓN. Sistemas Numéricos Posicionales. Números BINARIOS, OCTALES y HEXADECIMALES Conversiones entre sistemas numéricos posicionales. Operaciones Aritméticas con números no decimales		
II	REPRESENTACIÓN DE LA INFORMACIÓN. GENERALIDADES. Sistemas de procesamiento de la información. Tipos de datos. Constantes y variables. Expresiones: (Aritméticas, Relacionales, Lógicas, de Caracteres). Asignaciones. EJERCICIOS		
III	DISEÑO DE ALGORITMOS. PASOS EN LA RESOLUCIÓN DE UN PROBLEMA. Definición y Análisis, Diseño del algoritmo, Codificación y Validación MODELOS de REPRESENTACIÓN de ALGORITMOS. Diagramas de Flujo y Seudocódigo. DIAGRAMAS DE FLUJO. y SEUDOCÓDIGO Orientado al Lenguaje C#.Net de: ESTRUCTURAS BÁSICAS DE ALGORITMOS. • Estructuras selectivas. BIFURCACIONES. (Salida		

	simple, Salida doble y Salida múltiple. <ul style="list-style-type: none"> Estructuras repetitivas. BUCLES (Condición al final, Condición al inicio, Conociendo el número de iteraciones, Contadores y acumuladores) EJERCICIOS		
--	---	--	--

BIBLIOGRAFÍA RECOMENDADA

BIBLIOGRAFÍA BÁSICA

Título: "FUNDAMENTOS de INFORMÁTICA y PROGRAMACIÓN"
Autor: Martin, Gregorio & Toledo, Francisco & Cerveron, Vicente
Editorial:

Título: "ALGORITMOS Y ESTRUCTURA DE DATOS"
Autor: Joyanes Aguilar, Luis
Editorial: McGraw Hill

Título: "C. ALGORITMOS, PROGRAMACIÓN Y ESTRUCTURAS DE DATOS"
Autor: Sánchez García, Lucas Castillo Sanz, Andrés Zahonero, Ignacio Joyanes, Luis
Editorial: McGraw Hill/Interamericana de España.

BIBLIOGRAFÍA COMPLEMENTARIA

Título: "EJEMPLOS RESUELTOS DE DIAGRAMAS DE FLUJO Y CODIFICACIÓN"
Autor: Varios.
Editorial: UAGRM (Folleto)

Título: "ALGORITMOS RESUELTOS" (Estructuras Repetitivas)
Autor: Web
Editorial: programandoenc.webcindario.com

Título: "PROBLEMAS RESUELTOS CON DIAGRAMAS DE FLUJO"
Autor: Web
Editorial: www.monografias.com

Título: "MICROSOFT VISUAL C# STEP BY STEP".
Autor: Jhon Sharp & Jon Jagger
Editorial: Microsoft ®

**PLANIFICACIÓN DE ACTIVIDADES DEL MÓDULO
INTRODUCCIÓN a la PROGRAMACIÓN**

No.	FECHA	UNIDADES	ACTIVIDAD	ORIENTACIÓN
	Clase 1	TEMA-1	SISTEMAS DE NUMERACIÓN. Sistemas Numéricos Posicionales. SNP Números Binarios, Octales y Hexadecimales. Conversiones entre sistemas numéricos posicionales. SNP	
	Clase 2		Conversiones entre SNP Operaciones Aritméticas con números no decimales. Suma. Ejercicios	Se orienta el TC no 1. de Conversiones entre SNP
	Clase 3	TEMA-2	Operaciones Aritméticas con números no decimales. Resta. Ejercicios REPRESENTACIÓN DE LA INFORMACIÓN. GENERALIDADES. Sistemas de procesamiento de la información. Tipos de datos. Constantes y variables. Expresiones: Aritméticas. Ejercicios	Se recibe el TC no 1 Se orienta el TC no 2. de Operaciones Aritméticas con SNP
	Clase 4		Expresiones: Relacionales y Lógicas. Tablas de Verdad Expresiones de Caracteres Asignaciones. Ejercicios	Se recibe el TC no 2
	Clase 5		Representación algorítmica de expresiones algebraicas. EJERCICIOS.	
	Clase 6		EJERCICIOS	Se orienta el TC no 3. de Expresiones y Asignaciones
	Clase 7		1 PARCIAL Temas 1 y 2	Se recibe el TC no 3 → 15 puntos
	Clase 8	TEMA-3	TEMA-3 DISEÑO DE ALGORITMOS. PASOS EN LA RESOLUCIÓN DE UN PROBLEMA. Definición y Análisis, Diseño del algoritmo, Codificación y Validación. DIAGRAMAS DE FLUJO y SEUDOCÓDIGO (en C#) de: ESTRUCTURAS BÁSICAS DE ALGORITMOS. Est. selectivas. BIFURCACIONES.(Salida simple, doble y múltiple).	
	Clase 9		EJEMPLOS y Ejercicios de Estructuras Selectivas	
	Clase 10		EJEMPLOS y Ejercicios de Estructuras Selectivas	
	Clase 11		DIAGRAMAS DE FLUJO y SEUDOCÓDIGO (en C#) de: Estructuras Repetitivas BUCLES (Condición al final, Condición al inicio, Y Conociendo el número de iteraciones,	
	Clase 12		Contadores y acumuladores. EJEMPLOS y	

			EJERCICIOS	
	Clase 13		EJEMPLOS y Ejercicios de Estructuras Repetitivas	
	Clase 14		EJEMPLOS y Ejercicios de Estructuras Repetitivas	Se orienta el TC no 4. de Algoritmos
	Clase 15		2 PARCIAL Tema 3	Se recibe el TC no 4 → 20 pts
	Clase 16		EJEMPLOS y Ejercicios de Algoritmos de Estructuras Selectivas y Repetitivas de problemas complejos	
	Clase 17		EJEMPLOS y Ejercicios de Algoritmos de Estructuras Selectivas y Repetitivas de problemas complejos	
	Clase 18		EJEMPLOS y Ejercicios de Algoritmos de Estructuras Selectivas y Repetitivas de problemas complejos	
	Clase 19		EXAMEN FINAL	→ 20 pts
	Clase 20		Discusión de CASOS de ESTUDIO	→ 30 pts

TEMA-1 SISTEMAS DE NUMERACIÓN.

OBJETIVOS ESPECÍFICOS:

Al terminar el tema el alumno debe:

- Conocer a cabalidad que es un sistema numérico posicional.
- Identificar los sistemas Binarios, Octales y Hexadecimales.
- Dominar la conversión entre los sistemas Binarios, Octales y Hexadecimales.
- Dominar las operaciones aritméticas de suma y resta en los sistemas Binarios, Octales y Hexadecimales

CONTENIDO:

1.1 Sistemas Numéricos Posicionales

- Definición
- Representación del sistema decimal según suma de potencias

1.2 El Sistema Binario

- Representación del sistema binario como suma de potencias de base 2
- Representación gráfica de un byte y su relación del sistema binario a nivel de bit

1.3 Los Sistemas Octal y Hexadecimal

- Tabla de números decimales, binario, octal y hexadecimal

1.4 Conversiones entre Sistemas Numéricos Posicionales

- Cualquier base a decimal
- Base decimal a otra base
- Bases Octal y Hexadecimal a Binaria y viceversa
- Base Octal a Hexadecimal y viceversa

1.5 Operaciones Aritméticas con Números no Decimales

- Suma y resta de números binarios, octales y hexadecimales.

RESUMEN

*En capítulo se analizan los sistemas numéricos posicionales **SNP** que son aquellos en los cuales la posición de los dígitos que componen al número tienen un “peso asociado” en el valor del mismo. Estos números se pueden representar como una suma de potencias donde cada sumando es un dígito multiplicado por la base o raíz del número elevado a una potencia acorde a la posición del mismo.*

*Como bien los circuitos digitales tienen señales que normalmente están en una de dos condiciones: alto-bajo, cargado-descargado, activado-desactivado, estos representan a bits que toman los valores **0** y **1***

Aunque las bases octales y hexadecimales no se emplean frecuentemente éstas son convenientes para números con múltiples bits en un sistema digital y por ello son motivo de estudio.


*El sistema octal necesita **8 dígitos** por lo que emplea los dígitos de **0** a **7**, esto implica que se necesitan 3 bits para poder representar su máximo valor (7)*

*El sistema hexadecimal necesita **16 dígitos** por lo que complementa los dígitos decimales **0** a **9** con las letras **A** a la **F**, de igual forma se necesitan 4 bits para la representación de su máximo valor ($F=15$)*


Las conversiones entre sistemas de distintas bases son a veces necesarias para el trabajo de los sistemas informáticos, luego se establecen las reglas para las conversiones entre los distintos sistemas numéricos vistos en el tema.

*El capítulo termina haciendo referencia a la ejercitación de las operaciones aritméticas de **SNP** de bases no decimales.*

Definición de SNP

- El **sistema numérico tradicional** que se estudia en la escuela y se emplea diariamente **se conoce como sistema numérico posicional (SNP)**.
- Este sistema se representa por una **cadena de dígitos**, donde **cada posición del dígito tiene un "peso asociado"**. Ejemplo 

$$1734 = 1*1000 + 7*100 + 3*10 + 4*1$$


Note que **cada peso es una potencia de 10** según la **posición del dígito**, ya que: 

$$1734 = 1*10^3 + 7*10^2 + 3*10^1 + 4*10^0$$

Definición de SNP

- La introducción del **punto decimal** permite el uso de **potencias de 10 tanto positivas como negativas**. Ejemplo

$$5185.68 = 5*1000 + 1*100 + 8*10 + 5*1 + 6*0.1 + 8*0.01$$

Representando el numero como **potencias de 10** se tendrá: 

$$5185.68 = 5*10^3 + 1*10^2 + 8*10^1 + 5*10^0 + 6*10^{-1} + 8*10^{-2}$$

- En la expresión anterior la cifra **10** se conoce como la **base o raíz del Sistema Numérico**
- La raíz de los **SNP** debe ser un **valor entero r**
- Un **dígito en la posición i** tiene un **peso rⁱ**

Definición de SNP

- El **valor del número (D)** es la **sumatoria de los dígitos multiplicados por su correspondiente potencia de la raíz**, luego:

$$D = \sum_{i=-n}^{p-1} d_i * r^i$$


donde: **r** = base del número en cuestión
p = dígitos a la izquierda del punto
n = dígitos a la derecha del punto
i = posición del dígito dentro del numero
d_i = valor del dígito en la posición i

Definición de SNP

- Si se exceptúan los posibles ceros anteriores y posteriores la representación de un número en un **SNP es ÚNICA**
- El dígito que está **más a la izquierda** se llama dígito de orden superior o **más significativo** y el de **más a la derecha** es el dígito de orden inferior o **menos significativo**.

#s Binarios, Octales y Hexadecimales

- Decimales**
Dígitos válidos **0 .. 9**, base o raíz del sistema **r = 10**
- Binarios**
Dígitos válidos **0 y 1**, base o raíz del sistema **r = 2**
- Octales**
Dígitos válidos **0 .. 7**, base o raíz del sistema **r = 8**
- Hexadecimales**
Dígitos válidos **0 .. 9 y A .. F**, raíz del sistema **r = 16**
Las letras **A a la F** representan los dígitos **10 al 15**, según **A=10, B=11, C=12, D=13, E=14 y F=15**

Conversiones: Grupo I

- de cualquier base a la base Decimal

Empleo de la Fórmula $D = \sum d_i * r^i$

Ejemplos:

a) **binario a decimal**

$$10011_2 = 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0$$

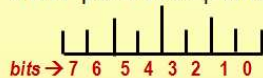
$$= 1*16 + 1*2 + 1*1 = 19_{10}$$

$$100010_2 = 1*2^5 + 1*2^1 = 34_{10}$$

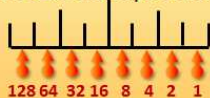
Note que en el segundo ejemplo solo se han considerado los **dígitos con valor 1**, ya que los de **valor 0 no aportan** al valor final.

Conversiones: Grupo I $D = \sum d_i \cdot r^i$

- Otra forma de considerar la conversión de binario a decimal es con la **representación grafica** de un **byte**. Como se conoce un byte está compuesto por 8 **bits*** por lo su esquema sería como el que sigue:



- Luego cada **bit** con el valor 1 dentro del byte aportará un valor equivalente a la **base 2** elevado al índice de dicha posición, luego se tendrían:



* la palabra **bit** es la contracción de **binary digit** (dígito binario)

Dibujado por Elican - Descartolite.com

Conversiones: Grupo I $D = \sum d_i \cdot r^i$

Analizando los ejemplos anteriores, se tendrá:

$$\begin{array}{cccccccc} & & & 1 & 0 & 0 & 1 & 1 \\ \hline & & & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = 16 + 2 + 1 = 19_{10}$$

$$\begin{array}{cccccccc} & & & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline & & & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = 32 + 2 = 34_{10}$$

$$\begin{aligned} 101.001_2 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 0 \cdot 0.5 + 0 \cdot 0.25 + 1 \cdot 0.125 \\ &= 5.125_{10} \end{aligned}$$

Dibujado

Conversiones: Grupo I $D = \sum d_i \cdot r^i$

b) octal a decimal

$$\begin{aligned} 1357_8 &= 1 \cdot 8^3 + 3 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 \\ &= 1 \cdot 512 + 3 \cdot 64 + 5 \cdot 8 + 7 \cdot 1 \\ &= 512 + 192 + 40 + 7 = 751_{10} \end{aligned}$$

$$\begin{aligned} 2046.17_8 &= 2 \cdot 8^3 + 4 \cdot 8^2 + 6 \cdot 8^1 + 1 \cdot 8^0 + 7 \cdot 8^{-1} + 7 \cdot 8^{-2} \\ &= 2 \cdot 512 + 4 \cdot 64 + 6 \cdot 8 + 1 \cdot 1 + 7 \cdot \frac{1}{8} + 7 \cdot \frac{1}{64} \\ &= 1024 + 256 + 48 + 1 + 0.875 + 0.109375 \\ &= 1330 + 0.984375 = 1330.984375_{10} \end{aligned}$$

Dibujado por Elican

Conversiones: Grupo I $D = \sum d_i \cdot r^i$

c) hexadecimal a decimal

$$\begin{aligned} 1CE8_{16} &= 1 \cdot 16^3 + 12 \cdot 16^2 + 14 \cdot 16^1 + 8 \cdot 16^0 \\ &= 1 \cdot 4096 + 12 \cdot 256 + 14 \cdot 16 + 8 \cdot 1 \\ &= 4096 + 3072 + 224 + 8 = 7400_{10} \end{aligned}$$

$$\begin{aligned} 9F.46C_{16} &= 9 \cdot 16^1 + 15 \cdot 16^0 + 4 \cdot 16^{-1} + 6 \cdot 16^{-2} + 12 \cdot 16^{-3} \\ &= 9 \cdot 16 + 15 \cdot 1 + 4 \cdot \frac{1}{16} + 6 \cdot \frac{1}{256} + 12 \cdot \frac{1}{4096} \\ &= 9 \cdot 16 + 15 \cdot 1 + \frac{4}{16} + \frac{6}{256} + \frac{12}{4096} \\ &= 144 + 15 + 0.25 + 0.0234375 + 0.0029296875 \\ &= 159 + 0.2763671875 = 159.2763671875_{10} \end{aligned}$$

Dibujado

Conversiones: Grupo II MDS

- de la base Decimal a cualquier base

Cuando se tiene una conversión de un número en base 10 a otra base se usa el **método de las divisiones sucesivas**, es decir:

Se realizan divisiones del número entre la base o raíz a la que se quiere convertir hasta obtener un cociente igual a cero, siendo los residuos de estas divisiones ordenado en forma inversa, el resultado

Dibujado por Elican

Conversiones: Grupo II MDS

a) decimal a binario

$$179_{10} \text{ a binario} = 10110011_2$$

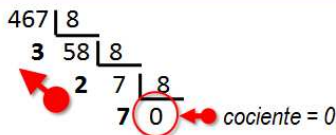
$$\begin{array}{r} 179 \overline{) 2} \\ 1 \ 89 \overline{) 2} \\ 1 \ 44 \overline{) 2} \\ 0 \ 22 \overline{) 2} \\ 0 \ 11 \overline{) 2} \\ 1 \ 5 \overline{) 2} \\ 1 \ 2 \overline{) 2} \\ 0 \ 1 \overline{) 2} \\ 1 \ 0 \overline{) 2} \end{array}$$

cociente = 0

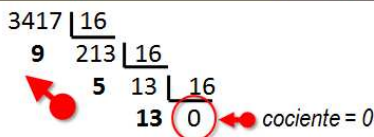
Dibujado

Conversiones: Grupo II MDS

b) decimal a octal 467_{10} a octal = 723_8



c) decimal a hexadecimal 3417_{10} a hexadecimal = $D59_{16}$



Conversiones: Grupo III

■ de la base octal y hexadecimal a binario y viceversa

Mediante sustitución de cada dígito octal o hexadecimal por su correspondiente valor binario

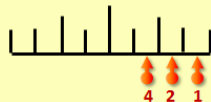
a) De octal a binario

Como una cadena de 3 bits puede tener 8 combinaciones diferentes cada cadena de 3 bits puede representar de una manera única un dígito octal.

Elaborado por: Oscar - Duany OTBNA

Conversiones: Grupo III

En otras palabras el máximo valor de dígito octal es el 7 el cual puede representarse en 3 dígitos binarios, de la siguiente manera:



Ya que al tercer bit le corresponde el valor 4, al segundo 2 y al primer bit el valor 1 que sumados darán el valor 7

Ejemplos:

$$1357_8 = 001\ 011\ 101\ 111_2$$

$$2046.17_8 = 010\ 000\ 100\ 110.001\ 111_2$$

Conversiones: Grupo III

b) De binario a octal

Es muy sencillo convertir un número binario a octal, esto se logra separando, de derecha a izquierda los dígitos a la izquierda del punto binario y de izquierda a derecha los dígitos a la derecha del punto binario, en grupos de 3 bits y reemplazarlos por sus correspondientes valores octales. Ejemplos:

$$100011001110_2 = 100\ 011\ 001\ 110_2 = 4316_8$$

$$111011011101001_2 = 011\ 101\ 101\ 110\ 101\ 001_2 = 355651_8$$

$$10.1011001011_2 = 010\ 101\ 100\ 101\ 100_2 = 2.5454_8$$

Conversiones: Grupo III

c) De hexadecimal a binario

De manera semejante al caso de los números octales una cadena de 4 bits puede representar de una manera única un dígito hexadecimal.

En otras palabras el máximo valor de dígito hexadecimal es 15 (F) el cual puede representarse en 4 dígitos binarios, como:



Ya que al cuarto bit le corresponde el valor 8 al tercer bit el valor 4, al segundo 2 y al primer bit el valor 1 que sumados darán el valor 15

Conversiones: Grupo III

c) De hexadecimal a binario

Ejemplos:

$$BEAD_{16} = 1011\ 1110\ 1010\ 1101_2$$

$$9F.46C_{16} = 1001\ 1111.0100\ 0110\ 1100_2$$

d) De binario a hexadecimal

De igual forma que en el caso de los números octales pero ahora considerando grupos de 4 bits puede convertirse de binario a hexadecimal.

Es decir se hacen grupos de 4 bits de derecha a izquierda a la izquierda del punto binario y de izquierda a derecha para los dígitos a la derecha del punto binario

Conversiones: Grupo III

c) De binario a hexadecimal

$$100011001110_2 = 1000\ 1100\ 1110_2 = 8CE_{16}$$

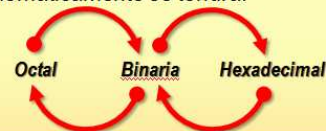
$$11101101110101001_2 = 0001\ 1101\ 1011\ 1010\ 1001_2 = 1DBA9_{16}$$

$$10.1011001011_2 = 0010\ 1011\ 0010\ 1100_2 = 2.B2C_{16}$$

Conversiones: Grupo IV

de la base octal a hexadecimal y viceversa

Para este caso la conversión se realiza **utilizando una base intermedia** (como por ejemplo la binaria), esquemáticamente se tendrá:



Ejemplos:

$$1357_8 = 0010\ 1110\ 1111_2 = 2EF_{16}$$

$$2046.17_8 = 0100\ 0010\ 0110\ 0011\ 1100_2 = 426.1C_{16}$$

Suma de Binarios, Octales y Hexadecimales

Suma decimal

Ejemplo de suma decimal y forma en que se producen los acarreos.

acarreo \rightarrow

1	1	0	14	10	4	10	4	10
3	8	5	4	1	0	1	4	0
+	1	9						
4	0	4						

Nótese que al ser la suma de la columna en cuestión mayor que el valor de la base se divide entre ésta última, siendo el **resto de dicha división el resultado de la columna y el cociente el acarreo**.

Suma de SNP: Metodología

La metodología a seguir será (para cualquier SNP) la de **dividir la suma de los números de la columna en cuestión entre la base del SNP** donde el **residuo de esta división será el resultado de la columna y el cociente el acarreo**.

Suma de números Binarios

Suma binaria

acarreo \rightarrow <101111000>

X	190	10111110 ₂	2+4+8+16+32+128 = 190
Y	+ 141	10001101 ₂	1+4+8+128 = 141
	331	101001011 ₂	1+2+8+64+256 = 331

acarreo \rightarrow <001011000>

X	173	10101101 ₂	1+4+8+32+128 = 173
Y	+ 44	00101100 ₂	4+8+32 = 44
	217	11011001 ₂	1+8+16+64+128 = 217

Suma de números Octales y Hexadecimales

Suma octal

acarreo \rightarrow <01010>

X	727	1327 ₈	1*8 ³ +3*8 ² +2*8 ¹ +7*8 ⁰ = 512+192+16+7 = 727
Y	+ 389	605 ₈	6*8 ² +0*8 ¹ +5*8 ⁰ = 384+0+5 = 389
	1116	2134 ₈	2*8 ³ +1*8 ² +3*8 ¹ +4*8 ⁰ = 1024+64+24+4 = 1116

Suma hexadecimal

acarreo \rightarrow <01010>

X	51174	C7E6 ₁₆	12*16 ³ +7*16 ² +14*16 ¹ +6*16 ⁰ = 51174
Y	+ 6585	19B9 ₁₆	1*16 ³ +9*16 ² +11*16 ¹ +9*16 ⁰ = 6585
	57759	E19F ₁₆	14*16 ³ +1*16 ² +9*16 ¹ +15*16 ⁰ = 57759

EJERCICIOS RESUELTOS

CONVERSIONES

Convertir de binario a decimal los siguientes números:

$$10011_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ = 1 \cdot 16 + 1 \cdot 2 + 1 \cdot 1 = 19_{10}$$

$$100010_2 = 1 \cdot 2^5 + 1 \cdot 2^1 = 34_{10}$$

$$101.001_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ = 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 0 \cdot 0,5 + 0 \cdot 0,25 + 1 \cdot 0,125 = 5,125_{10}$$

Convertir:

a) 179_{10} binario

$$\begin{array}{r} 179 / 2 = 89 \text{ residuo } 1 \text{ (lsb)} \\ \quad 89 / 2 = 44 \text{ residuo } 1 \\ \quad \quad 44 / 2 = 22 \text{ residuo } 0 \\ \quad \quad \quad 22 / 2 = 11 \text{ residuo } 0 \\ \quad \quad \quad \quad 11 / 2 = 5 \text{ residuo } 1 \\ \quad \quad \quad \quad \quad 5 / 2 = 2 \text{ residuo } 1 \\ \quad \quad \quad \quad \quad \quad 2 / 2 = 1 \text{ residuo } 0 \\ \quad \quad \quad \quad \quad \quad \quad 1 / 2 = 0 \text{ residuo } 1 \text{ (msb)} \end{array}$$

luego:

$$179_{10} = 10110011_2$$

b) 467_{10} a octal

$$\begin{array}{r} 467 / 8 = 58 \text{ residuo } 3 \text{ (lsb)} \\ \quad 58 / 8 = 7 \text{ residuo } 2 \\ \quad \quad 7 / 8 = 0 \text{ residuo } 7 \text{ (msb)} \end{array}$$

luego:

$$467_{10} = 723_8$$

c) 3417_{10} a hexadecimal

$$\begin{array}{r} 3417 / 16 = 213 \text{ residuo } 9 \text{ (lsb)} \\ \quad 213 / 16 = 13 \text{ residuo } 5 \\ \quad \quad 13 / 16 = 0 \text{ residuo } 13 \text{ (msb)} \end{array}$$

luego:

$$3417_{10} = D59_{16}$$

Convertir de binario a octal los siguientes números:

$$100011001110_2 = 100 \ 011 \ 001 \ 110_2 = 4316_8$$

$$11101101110101001_2 = 011 \ 101 \ 101 \ 110 \ 101 \ 001_2 = 355651_8$$

$$10.1011001011_2 = 010 \ . \ 101 \ 100 \ 101 \ 100 = 2.5454_8$$

Convertir de binario a hexadecimal los siguientes números:

$$100011001110_2 = 1000 \ 1100 \ 1110_2 = 8CE_{16}$$

$$11101101110101001_2 = 1 \ 1101 \ 1011 \ 1010 \ 1001_2 = 1DBA9_{16}$$

$$10.1011001011_2 = 0010 \ . \ 1011 \ 0010 \ 1100 = 2.B2C_{16}$$

Convertir de octal a binario los siguientes números:

$$1357_8 = 001 \ 011 \ 101 \ 111_2$$

$$2046.17_8 = 010 \ 000 \ 100 \ 110 \ . \ 001 \ 111_2$$

Convertir de hexadecimal a binario los siguientes números:

$$BEAD_{16} = 1011 \ 1110 \ 1010 \ 1101_2$$

$$9F.46C_{16} = 1001 \ 1111 \ . \ 0100 \ 0110 \ 1100_2$$

Comprobar los resultados de las conversiones siguientes:

Convertir de Binario a Octal y Hexadecimal los siguientes valores:

$$010011101110_2 \quad (2356_8 \quad 4EE_{16})$$

$$111011101110_2 \quad (7356_8 \quad EEE_{16})$$

$$1011011101111_2 \quad (13357_8 \quad 16EF_{16})$$

111101.110110100₂ (75.664₈ 6D.DA₁₆)

Convertir de Octal y Hexadecimal a Binario los siguientes valores:

257₈ (010 101 111₂)
 3521₁₆ (0011 0101 0010 0001₂)
 475₁₆ (0100 0111 0101₂)
 2720₈ (010 111 010 000₂)
 3630₁₆ (0011 0110 0011 0000₂)

Realizar las conversiones de Octal a Hexadecimal y viceversa siguientes:

1234₈ (001 010 011 100₂ 0010 1001 1100₂ 29C₁₆)
 CODE₁₆ (1100 0000 1101 1110₂ 001 100 000 011 011 110₂ 140336₈)

OPERACIONES ARITMÉTICAS BÁSICAS

SUMA

C acarreo	<u><011111110></u> X 127	C acarreo	<u><000000000></u>	X 170	10101010 ₂
Y	<u>63</u> 190	Y	<u>85</u> 255		
	00111111 ₂ <u>10111110₂</u>		01010101 ₂ <u>11111111₂</u>		

C acarreo	<u><01010></u> X 727	1. 8 ³ +3 . 8 ² +2 . 8 ¹ +7 . 8 ⁰ = 512+192+16+7 = 727
Y	<u>389</u> 1116	6 . 8 ² +0 . 8 ¹ +5 . 8 ⁰ = 384+0+5 = 389
	605 ₈ <u>2134₈</u>	2 . 8 ³ +1 . 8 ² +3 . 8 ¹ +4 . 8 ⁰ = 1024+64+24+4 = 1116

C acarreo	<u><01100></u> X 51174	12 . 16 ³ +7 . 16 ² +14 . 16 ¹ +6 . 16 ⁰ =49152+1792+224+6 = 51174
Y	<u>6585</u> 57759	1 . 16 ³ +9 . 16 ² +11 . 16 ¹ +9 . 16 ⁰ = 4096+2304+176+9 = 6585
	19B9 ₁₆ <u>E19F₁₆</u>	57759 / 16 = 3609 res 15 (lsb) / 16 = 225 res 9 / 16 = 14 res 1 / 16 = 0 res 14 (msd)

RESTA

B préstamo	<u><00000000></u> X 127	B préstamo	<u><10101010></u>	X 170	10101010 ₂
Y	<u>63</u> 64	Y	<u>85</u> 85		
	00111111 ₂ <u>01000000₂</u>		01010101 ₂ <u>01010101₂</u>		

B préstamo	<u><1000></u> X 727	1. 8 ³ +3 . 8 ² +2 . 8 ¹ +7 . 8 ⁰ = 512+192+16+7 = 727
Y	<u>389</u> 338	6 . 8 ² +0 . 8 ¹ +5 . 8 ⁰ = 384+0+5 = 389
	605 ₈ <u>522₈</u>	5 . 8 ² +2 . 8 ¹ +2 . 8 ⁰ = 320+16+2 = 338

B préstamo	<u><1010></u> X 51174	12 . 16 ³ +7 . 16 ² +14 . 16 ¹ +6 . 16 ⁰ =49152+1792+224+6 = 51174
Y	<u>6585</u> 44589	1 . 16 ³ +9 . 16 ² +11 . 16 ¹ +9 . 16 ⁰ = 4096+2304+176+9 = 6585
	19B9 ₁₆ <u>AE2D₁₆</u>	10 . 16 ³ +14 . 16 ² +2 . 16 ¹ +13 . 16 ⁰ =40960+3584+32+13 = 44589

EJERCICIOS PROPUESTOS

CONVERSIONES

Convertir de binario a decimal los siguientes números:

a) 101_2 b) 0111_2 c) 10101111_2 d) 1000111010101_2

Convertir de decimal a binario los siguientes números:

a) 278 b) 936 c) 1897 d) 5863

Convertir de decimal a hexadecimal los siguientes números:

a) 15 b) 59 c) 268 d) 873

Convertir de hexadecimal a decimal los siguientes números:

a) $1A_{16}$ b) $D8_{16}$ c) $5C2_{16}$ d) $4FB_{16}$

Convertir de octal a decimal los siguientes números:

a) 64_8 b) 105_8 c) 1025_8 d) 4530_8

Convertir de decimal a octal los siguientes números:

a) 43 b) 80 c) 236 d) 4586

Convertir de binario a octal los siguientes números:

a) 11010101_2 b) 11110011_2 c) 100001111011_2 d) 1100001111101_2

OPERACIONES ARITMÉTICAS BÁSICAS

Realizar las siguientes sumas binarias:

a) $101_2 + 10_2$ b) $110_2 + 11_2$

c) $1001_2 + 1110_2$ d) $10100_2 + 11011_2$

Realizar las siguientes restas binarias:

a) $1011_2 - 100_2$ b) $101110110_2 - 10000_2$

c) $1000001000_2 - 1110011_2$ d) $1101000010_2 - 111111_2$

TEMA-2 REPRESENTACIÓN DE LA INFORMACIÓN.

OBJETIVOS ESPECÍFICOS

Al terminar el tema el alumno debe:

- Identificar los conceptos generales que se tomarán en cuenta a lo largo de la materia para la resolución de problemas como son variables, constantes, tipos de datos, operadores y expresiones
- Identificar las operaciones de asignación y las de entrada – salida como operaciones básicas de cualquier programa de computación

CONTENIDO:

2.1 Generalidades.

- Sistema de Procesamiento de la Información

2.2 Representación de la Información

- Tipos de datos.
- Constantes y variables.
- Expresiones: (Aritméticas, Relacionales, Lógicas, de Caracteres).
- Asignaciones.

RESUMEN:

En el tema se plantea el mecanismo de procesamiento de la información (datos, programa, salida), como sistema de información, considerándose fundamentalmente los tipos de datos numéricos (enteros y reales), booleanos o lógicos y los de carácter – cadena.

Los programas de computación contienen ciertos valores que no cambian durante la ejecución del programa y otros que pueden cambiar durante la ejecución del mismo, a los primeros se les denominan constantes mientras que a los últimos se les conoce como variables.

*Las expresiones en su forma mas simple están definidas según: **operando – operador – operando** siendo, generalmente, los operandos, variables y constantes y los operadores símbolos que indican la operación (aritmética o lógica) a realizar con los operandos.*

Cuando las expresiones son simples no existe duda en cuanto a la operación a realizar, pero cuando son compuestas, es decir, cuando intervienen mas de dos operadores, debe existir un orden o precedencia para realizar las operaciones señaladas.

Cuando una constante, variable o resultado de una expresión se asigna a una variable el valor contenido en ésta es reemplazado por el valor asignado conociéndose esta operación como operación de asignación.

Un algoritmo no es mas que una serie de instrucciones o sentencias que especifican la secuencia de pasos a seguir en la resolución de un problema, estando éste último regido por, al menos, de una serie de pasos como la definición y análisis del problema, el desarrollo del algoritmo, la codificación o representación mediante un lenguaje de programación y la validación o corrección de la representación del problema hasta alcanzar los resultados esperados.

Finalmente en el tema se estudia la representación algorítmica de diversas expresiones algebraicas.

Esquema de Procesamiento de la Info

- La gran mayoría de los programas de computación posee el siguiente esquema de procesamiento de la Información.



Dibujado por Elzabete - Desean UTEDSA

Pasos en la Resolución de Problemas

La resolución de un problema exige al menos los siguientes pasos:

- **Análisis** (y definición) del problema
- **Diseño del algoritmo**

Se puede definir como **algoritmo** al **conjunto de instrucciones o sentencias** que especifican la **secuencia de pasos** para resolver un determinado problema.

- **Codificación** (transformación) del algoritmo a un lenguaje de programación
- **Validación**

Dibujado por Elzabete - Desean

Tipos de Datos

Como se puede apreciar en el esquema de un sistema de la información éste comienza con el manejo de la información en forma de datos. **Los datos**, en forma simple, **pueden agruparse en varios tipos**, como por ejemplo:

- **Numéricos** (enteros y reales)
- **Lógicos** (booleanos)
- **Caracteres** (carácter y cadenas de caracteres)

Dibujado por Elzabete - Desean UTEDSA

Tipos de Datos: Numéricos

Representan el **conjunto de los valores numéricos**. Pueden ser **enteros** o **reales**

- **Enteros**: subconjunto finito de los números enteros. No poseen parte fraccionaria y pueden ser positivos y negativos. Ejemplos: **9, 18, 1340, -28**
- **Reales**: subconjunto finito de los números reales. Poseen parte fraccionaria y pueden ser positivos y negativos. Ejemplos: **0.08, -52.521, 3.141592653**

Para el empleo de números muy grandes o muy pequeños se emplea la notación exponencial. *Ejemplos:*

$$328\ 150\ 000\ 000\ 000 = 3.2815 \times 10^{14} = 3.2815E+14$$
$$0.00000000001532 = 1.532 \times 10^{-11} = 1.532E-11$$

Tipos de Datos: Lógicos

Representan valores booleanos, por lo que toman dos tipos de valores:

Verdadero = V Falso = F

Utilidad: para **representar alternativas** (si / no) a determinadas **condiciones**.
Por **ejemplo** si se pregunta si un valor entero es par la respuesta será **V** o **F** si el número es par o impar.

Dibujado por Elzabete - Desean UTEDSA

Tipos de Datos: de Caracteres

Pueden ser de dos tipos: **carácter** o **cadena**
El tipo **carácter** contiene un solo carácter encerrado entre apóstrofes (*para el lenguaje C#.Net*)
Una **cadena** de caracteres es una sucesión de caracteres delimitado por comillas.

La mayoría de las computadoras reconocen el juego de caracteres **ASCII** (*American Standard Character International Interchange*) que entre otros contiene los siguientes:

- **Alfabéticos** (letras **A..Z, a..z**)
- **Numéricos** (números **0..9**)
- **Especiales** (**+, -, *, /, :, <, >, \$,** etc)

Dibujado por Elzabete - Desean

❑ Constantes y Variables

- Los programas de computación contienen ciertos valores que **NO cambian durante la ejecución del programa**, estos valores se denominan **constantes**.
- De igual forma otros cambian durante la ejecución del programa, a éstos se les denominan **variables**.
- Debe tenerse presente que **una variable que es de un tipo solo puede tomar valores de ese tipo**. Es decir que si una variable es de **tipo entero** solo podrán asignársele **valores de tipo entero**, cualquier otro intento producirá un **"error de tipo"**

Elabórame por Elicabe - Docuam IT/ENSA

❑ Definición y Sintaxis

- Las expresiones son **combinaciones de constantes, variables, símbolos de operaciones, funciones especiales, etc.** *Ejemplo: $a + (b/3)^*$*
- En general una expresión consta de **operandos y operadores**. En la anterior expresión se tienen los **operandos: a, b, 3** y los **operadores: +, /, ***

La forma más simple de **representación** de una **expresión** es:

operando – operador – operando

Elabórame por Elicabe - Docuam IT/ENSA

❑ Expresiones Aritméticas

- Los operadores aritméticos en **C#.Net** son:
+ suma – resta * multiplicación / división % resto división entera
- Los operadores / y %
En **C#** la **división entera** se produce cuando **ambos operandos** son valores numéricos del **tipo entero**, si uno de ellos al menos es de **tipo real** el resultado será de **tipo real**.
El **operador %** indica el resto de la división entera de los operandos. Ejemplos:

15 / 6.0 = 2.5
15 / 6 = 2
15 % 6 = 3

+
-
*
/
%

Elabórame por Elicabe - Docuam IT/ENSA

❑ Expresiones Relacionales

- Las **expresiones relacionales** dan como resultados **valores lógicos** o booleanos (V o F).
- Los operadores de relación **permiten realizar comparaciones** de valores de tipo numérico o de carácter.
- Estos operadores **se usan para expresar condiciones** en los algoritmos.

Operador significado

<	menor que
>	mayor que
==	igual que
<=	menor o igual que
>=	mayor o igual que
!=	distinto de

>
<
==
!=
<=
>=
!=

Elabórame por Elicabe - Docuam IT/ENSA

❑ Expresiones Relacionales

El formato general para las comparaciones es:

expresión1 – operador relacional – expresión2

siendo el resultado de la comparación un valor **V** o **F**
Ejemplo: si A=4 y B=3 entonces **A > B** da **V**
mientras que: **(A – 2) < (B – 4)** da **F**

- ➔ **expresion1** y **expresion2** son expresiones **numéricas, de caracteres o cadenas**.

Nota. - Cuando se usan operadores de relación con operandos de tipo lógico se considera la condición falsa de valor menor que la verdadera así: **F < V** y de igual forma **V > F**
Se asocia (para muchos lenguajes) que **F = 0** y **V = 1**

Elabórame por Elicabe - Docuam IT/ENSA

❑ Expresiones Lógicas

- Estas expresiones al igual que las expresiones relacionales **dan como resultados valores lógicos o booleanos (V o F)**.
- Los principales operadores lógicos son: **no, y, o**
- El formato de estas expresiones es:

expresión1 – operador lógico – expresión2

- ➔ **expresión1** y **expresión2 TIENEN** que ser expresiones de booleanas, es decir, que devuelvan resultados de tipo **V** o **F**.

Elabórame por Elicabe - Docuam IT/ENSA

Operadores Lógicos



- Los operadores mas utilizados son:
! negación **&&** conjunción **||** disyunción
- El significado de estos operadores se puede apreciar en la siguiente tabla:

Operador	exp lógica	significado
no (!)	!exp1	Negación de exp1 Operador unario
y (&&)	exp1 && exp2	Conjunción de exp1 y exp2 Operador binario
o ()	exp1 exp2	Disyunción de exp1 y exp2 Operador binario

Tablas de Verdad – 1

- Negación:**
Tiene como significado **NO** o **FALSO**

exp	!exp
F	V
V	F

Tablas de Verdad – 2

- Conjunción:**
Usa el término de enlace “y” (operador C# &&)

exp1	exp2	exp1 && exp2
F	F	F
V	F	F
F	V	F
F	F	V

Nota: Si al menos uno de los operandos es **F** el resultado es **F**, solo si ambos son **V** el resultado es **V**
 Otra forma de considerarlo es como **producto lógico (*)** tomando valores de **F=0** y **V=1**

Tablas de Verdad – 3

- Disyunción Inclusiva:**
Usa el término de enlace “o” (operador C# ||)

exp1	exp2	exp1 exp2
F	F	F
V	F	V
F	V	V
F	F	V

Nota: Si al menos uno de los operandos es **V** el resultado es **V**, solo si ambos son **F** el resultado es **F**
 Otra forma de considerarlo es como **suma lógica (+)** tomando valores de **F=0** y **V=1**

Expresiones de Cadenas

- El único operador de cadenas es el operador **+** lo cual implica la **concatenación** de otra cadena o un carácter.

Ejemplos: 'A' + 'B' + 'C' “ABC”
 “Visual ” + “Studio” “Visual Studio”
 “C#” + ‘.’ + “NET” “C#.NET”

Reglas de Precedencias

- 5 + 3** expresión **simple** = un único operador
- 5 + 3 * 7 % 2** **compuesta** = mas de un operador debe atenderse a la tabla de prioridades

nivel	tipo	Descripción	Operador
1		Paréntesis (<i>agrupamiento</i>)	()
2	Lógica	Negación	! no
3	Aritmética	Multiplicación, División y Resto de la división entera	*, /, %
4		Suma y Resta	+, -
5	Relacional	Operadores relacionales	>, <, >=, <=
6			==, !=
7	Lógica	Conjunción y Disyunción	&& y, o

❑ La Operación de Asignación

- Modo de darle valores a una variable, se representa en el lenguaje algorítmico mediante \leftarrow

Formato: **nombre_var** \leftarrow **exp.**

donde **exp** puede ser una expresión, una variable o una constante. Ejemplo:

a \leftarrow **25** indica que se ha asignado el valor **25** a la variable **a**

- La operación de asignación **reemplaza el valor de la variable por el asignado**, ejemplo:

a \leftarrow **25**

x \leftarrow **4**

a \leftarrow **3**

x \leftarrow **48**

Los valores finales de **a** y **x** serán: **a=3** y **x=48**

Elaborado por Elvira - Duran UTEPA

❑ Exp. Algebraicas \rightarrow Algorítmicas

Exp. Algebraica	Significado	Exp. Algorítmica
$ x $	Valor absoluto de x	abs(x)
$\tan^{-1}(x)$	Angulo cuya tangente es x	atan(x)
sen(x), cos(x)	Seno y Coseno de x	sen(x), cos(x)
$\ln(x), \log_{10}(x)$	Logaritmos, natural y base 10 de x	ln(x), log10(x)
	Redondeo de x	redondeo(x)
	5.2 \rightarrow 5 5.7 \rightarrow 6 5.5 \rightarrow 6 6.5 \rightarrow 6	
x^p	x elevado a la p	pot(x,p)
	x ² \rightarrow cuadrado(x) x ³ \rightarrow cubo(x)	
$\sqrt[r]{x}$	raíz r de x	raizr(x)
	Truncar el valor de x	truncar(x)
	5.99 \rightarrow 5	

Elaborado por Elvira - Duran UTEPA

PREGUNTAS de AUTOCONTROL:

- ¿A que se denomina dato en programación?
- ¿Qué se entiende por algoritmo y como puede ser representado?
- ¿Qué características poseen las operaciones de entrada – salida en un programa?
- ¿Dónde pueden ser usadas las expresiones relacionales y lógicas en un programa?
- Diga, a su criterio, ¿que importancia revisten las operaciones de entrada – salida?
- Como Ud. conoce en una operación de asignación es posible asignar el contenido de una expresión a una variable, a lo que se denomina operación de asignación. ¿Podría esa misma variable encontrarse como un operando de la expresión? Explique.

EJEMPLOS RESUELTOS:

I Indicar el valor de cada una de las siguientes variables:

- a** \leftarrow **pot(2,3) + 9 / 5 - 6 * 8**
a \leftarrow **8 + 1.8 - 48**
a \leftarrow **-38.2**
- b** \leftarrow **4 * pot(3,2) / 8 + 5**
b \leftarrow **4 * 9 / 8 + 5**
b \leftarrow **36 / 8 + 5**
b \leftarrow **9.5**
- c** \leftarrow **pot(12,2) / (8 + 5)**
c \leftarrow **144 / 13**
c \leftarrow **11.07**
- d** \leftarrow **(3 * 3) - 9 / (5 - 6) * 8**
d \leftarrow **9 - 9 / (-1) * 8**
d \leftarrow **9 + 9 * 8**
d \leftarrow **81**
- e** \leftarrow **125 / pot(5,2) - 5**
e \leftarrow **125 / 25 - 5**
e \leftarrow **5 - 5**
e \leftarrow **0**

II Hallar el valor de la variable **z** en cada una de las expresiones:

Si:

- $a \leftarrow 6$
 $b \leftarrow a * 10 - 3$ ($b \leftarrow 6 * 10 - 3$ $b \leftarrow 57$)
 $c \leftarrow 7 + b / 3$ ($c \leftarrow 7 + 57 / 3$ $c \leftarrow 7 + 19$ $c \leftarrow 26$)
6. $z \leftarrow a + b \% c$
 $z \leftarrow 6 + 57 \% 26$
 $z \leftarrow 6 + 5$
 $z \leftarrow 11$
7. $z \leftarrow (a + b) / c$
 $z \leftarrow (6 + 57) / 26$
 $z \leftarrow 63 / 26$
 $z \leftarrow 2$
8. $z \leftarrow b / a + \text{pot}(c, 2)$
 $z \leftarrow 57 / 6 + 676$
 $z \leftarrow 9 + 676$
 $z \leftarrow 685$
9. $z \leftarrow a * b / (c - 6)$
 $z \leftarrow 6 * 57 / (26 - 6)$
 $z \leftarrow 6 * 57 / 20$
 $z \leftarrow 17$
10. $z \leftarrow (b + c) / a + (b + c) \% a$
 $z \leftarrow (57 + 26) / 6 + (57 + 26) \% 6$
 $z \leftarrow 83 / 6 + 83 \% 6 = 13 + 5$
 $z \leftarrow 18$

III Evaluar las expresiones algorítmicas siguientes:

11. **cuadrado(3) > abs(9.05)**
 $9 > 9.05$
F
12. **no(16 * 2 > pot(5,2)) && (24 % 5 / 2 == 2)**
 $\text{no}(32 > 25) \ \&\& \ (4 / 2 == 2)$
 $\text{no}(V) \ \&\& \ V$
F && V
F
13. **raiz2(625) == cuadrado(5)**
 $25 == 25$
V
14. **pot(4,2) + 9 > cuadrado(5)**
 $16 + 9 > 25$
 $25 > 25$
F
15. **((6 + pot(3,2)) > abs(15.1)) || (25 == (9 + pot(4,2)))**
 $((6 + 9) > 15.1) \ || \ (25 == (9 + 16))$
 $(15 > 15.1) \ || \ (25 == 25)$
F || V
V

EJERCICIOS PROPUESTOS

1. Indicar el valor de cada una de las siguientes variables:

a) $x \leftarrow (2 + 3) * 6$

b) $x \leftarrow 4^2 + 7 * 6$

c) $x \leftarrow (4^2 + 7) * 6$

d) $x \leftarrow 4^2 + (7 * 6)$

e) $y \leftarrow \frac{9}{5} + 7$

f) $y \leftarrow \frac{54}{3^3}$

g) $y \leftarrow \frac{126}{\frac{3}{21}}$

2. Indicar el valor de cada una de las siguientes variables:

a) $a \leftarrow 3$	$b \leftarrow 6$	$c \leftarrow 8$	$d \leftarrow a + b - c$
b) $a \leftarrow 2$	$b \leftarrow 10$	$c \leftarrow 7$	$d \leftarrow (a + b) \% c$
c) $w \leftarrow 9$	$x \leftarrow 2$	$y \leftarrow 3$	$z \leftarrow x + w / y$
d) $q \leftarrow 2$	$e \leftarrow 34$	$r \leftarrow 8$	$h \leftarrow ((e / q) \% q - r) * 3$
e) $q \leftarrow 2$	$e \leftarrow 34$	$r \leftarrow 8$	$h \leftarrow ((e+r) \% 7) * (q - r)$

3. Evaluar las siguientes expresiones:

a) $-4 * 7 + \text{pot}(2,3) / 4 - 5$

b) $(24 \% 5) / 2$

c) $\text{pot}(5,2) / (15 + 5 * 2)$

d) $12 + 3 * 7 + 5 * 4$

e) $(\text{pot}(4,2) + \text{pot}(2,2)) < (10 + 5 / 2)$

f) $(1.0 * 3.0) == \text{raiz2}(9)$

g) $\text{cuadrado}(4) >= \text{abs}(16.5)$

h) $(\text{pot}(5,2) > \text{pot}(2,5)) \|\| (\text{pot}(3,2) > \text{pot}(2,3))$

i) $\text{no}((24 \% 5) / 2 == 2)$

j) $\text{no}(\text{raiz2}(31 + 5) == 6) \&\& (0 < 1)$

4. Encontrar el valor de la variable **VALOR** después de la ejecución de las siguientes operaciones.

a) $\text{VALOR} \leftarrow 4.0 * 5$

b) $X \leftarrow 3.0$
 $Y \leftarrow 2.0$
 $\text{VALOR} \leftarrow X * Y - Y$

c) $\text{VALOR} \leftarrow 5$
 $X \leftarrow 3$
 $\text{VALOR} \leftarrow \text{VALOR} * X$

d) $\text{VALOR} \leftarrow (10 > 6) Y (10 == 6)$

e) $\text{VALOR} \leftarrow 2 * 2 + 5 * 5$
 $\text{VALOR} \leftarrow \text{VALOR} + 1$

5. Deducir el valor que tienen las variables A y B al final de la siguiente secuencia de instrucciones.

$A \leftarrow 4$
 $B \leftarrow 5$
 $B \leftarrow A + 1$
 $B \leftarrow B + 1$
 $A \leftarrow B$

TEMA-3 DISEÑO DE ALGORITMOS.

OBJETIVOS ESPECÍFICOS

Al terminar el tema el alumno debe:

- Desarrollar el pensamiento lógico y la capacidad de descomponer la solución de un problema en pasos ordenados con el fin de diseñar algoritmos sencillos
- Identificar las estructuras básicas de los algoritmos y aplicarlas adecuadamente en la resolución de problemas
- Representar gráficamente los algoritmos mediante diferentes métodos

CONTENIDO:

3.1 Pasos en la Resolución de un Problema.

- Definición y Análisis,
- Diseño del algoritmo,
- Codificación y
- Validación

3.2 Diagramas de Flujo.

- Símbolos gráficos y pseudocódigo orientado al lenguaje C#.Net

3.3 Estructuras Básicas de Algoritmos.

- Estructuras selectivas. **BIFURCACIONES.**
Salida simple,
Salida doble y
Salida múltiple.
- Estructuras repetitivas. **BUCLES**
Condición al final,
Condición al inicio,
Conociendo el número de iteraciones,
Contadores y acumuladores

RESUMEN:

Se detallan los pasos a seguir en la resolución de un problema, vistos en el tema anterior, siendo el paso del análisis desglosado en los datos de entrada, los resultados, las fórmulas necesarias si éstas existieran y la lógica a emplear, es decir, que herramientas de programación se emplearán y como para resolver el problema.

La definición del problema es un aspecto crucial pues el programador debe de estar muy claro en que es lo que se le está solicitando.

El análisis debe aprovecharse para definir mediante identificadores cuales son las variables que intervendrán en el problema.

Existen varios métodos para el desarrollo de un algoritmo como son los de pseudo código, los diagramas de flujo y los lenguajes de programación. El pseudocódigo es el planteamiento del problema mediante un lenguaje coloquial, los diagramas de flujo mediante símbolos gráficos y los lenguajes de programación mediante el código directo que interpretará la PC, éste último constituye la forma mas avanzada de plantear el algoritmo por lo que para ello el programador debe contar con suficiente experiencia. En el caso de los dos primeros métodos es necesaria la codificación, es decir, pasarlo a un lenguaje de programación determinado.

Se estudian las estructuras selectivas o bifurcaciones que son las empleadas para la consideración de variantes de acuerdo al comportamiento de una determina condición, de esta forma se plantean las bifurcaciones de salida simple cuando el resultado de la condición es verdadero o de salida doble cuando el flujo de la ejecución toma caminos en dependencia si la condición es verdadera o si es falsa, asimismo se plantea el caso de la bifurcación con salidas múltiples donde al evaluar una expresión o variable, el flujo seguirá acorde al valor tomado.

Un bucle o lazo es un segmento del programa que se repite un número de veces mientras se cumpla una condición determinada y donde en cada ciclo o iteración se comprueba la condición. Existen bucles en los

que la condición se encuentra al final y otros donde ésta está al inicio, así como pueden existir bucles donde la cantidad de iteraciones o repeticiones son conocidas de antemano.

Los procesos repetitivos son la base del uso de las computadoras. En estos procesos se necesita, frecuentemente, contar las repeticiones que se efectúan, esto es lo que se denomina un contador. Luego un contador es una variable cuyo valor se incrementa o disminuye en una cantidad constante en cada iteración, asimismo un acumulador o totalizador es una variable cuya misión es almacenar cantidades variables resultantes de sumas sucesivas.

Los contadores y los acumuladores necesitan ser inicializados con un valor antes de la ejecución del bucle, ya que las variables que lo identifican se encuentran en la expresión que se les asignará.

Las acciones de un algoritmo definidas en sus pasos se traducen en la codificación del programa como instrucciones o sentencias, luego el código consta de una serie de instrucciones o sentencias cada una especificando las operaciones que debe hacer la computadora.

❑ Pasos para Resolver Problemas

1 Análisis del Problema

En este punto es necesario atender de los aspectos siguientes;

- **Datos de entrada** o de inicio al problema. (generalmente brindados por el usuario)
- **Resultados de salida** que cumplen con la expectativa del problema
- **Fórmulas** a emplear. En caso que sean necesarios deben ser incluidas en el proceso de resolución.
- **Lógica del problema** o razonamiento que realiza el programador buscando una lógica que resuelva la problemática planteada.

❑ Pasos para Resolver Problemas

2 Desarrollo del Algoritmo

Debe definirse el algoritmo que se empleará en la representación del problema.

Los **modelos** en el planteamiento o desarrollo de los algoritmos son los siguientes:

- **Diagrama de Flujo** símbolos gráficos unidos entre sí mediante líneas de flujo
- **Seudocódigo** lenguaje coloquial
- **Código** escritura de forma directa en un lenguaje de programación

❑ Pasos para Resolver Problemas

3 Codificación

Traducción del algoritmo a un **lenguaje de programación**.

Válido para los dos primeros modelos de algoritmos, es decir, los diagramas de flujo y los pseudocódigos

Elaborado por Elzabete - Docentes UTFRSA

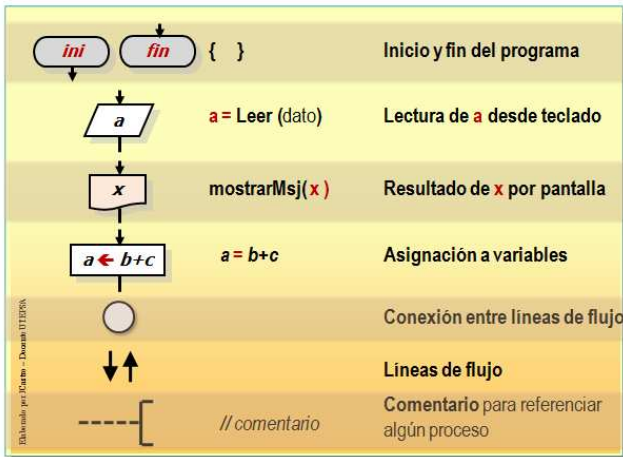
❑ Pasos para Resolver Problemas

4 Validación

- Análisis del algoritmo y **depuración de errores** hasta la obtención del resultado esperado.
- Para ello se procede a realizar una **prueba de escritorio**, o sea, resolver el problema a mano, el cual posteriormente se ejecutara en la PC y se comprobaran los resultados.

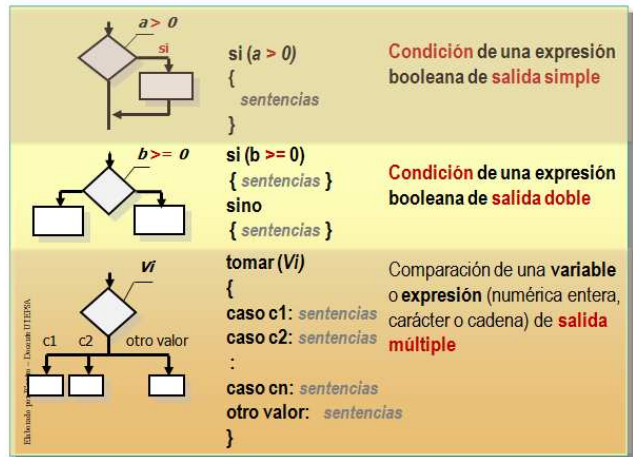
Elaborado por Elzabete - Docentes UTFRSA

DF y Seudocódigo en C#



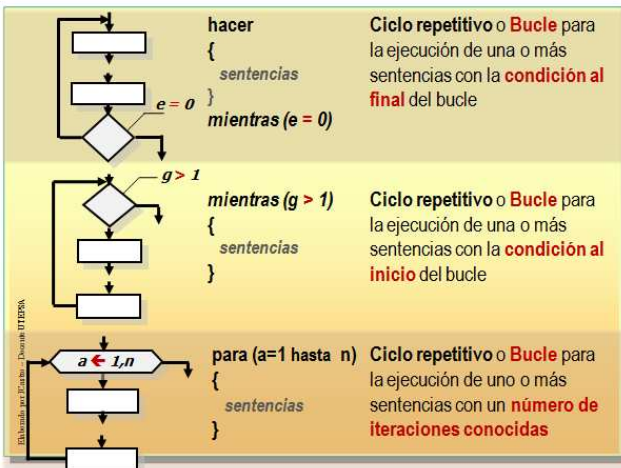
Elaborado por Elzabete - Docentes UTFRSA

DF y Seudocódigo en C#



Elaborado por Elzabete - Docentes UTFRSA

DF y Seudocódigo en C#



Elaborado por Elzabete - Docentes UTFRSA

Ejemplo

Calcular la **superficie** (área) y **longitud de la circunferencia** de un círculo de radio **r** dado por el usuario desde teclado.

1 Análisis del Problema

Datos: radio (**r**) teclado por el usuario

Resultados: área (**a**) del círculo de radio **r** y longitud (**L**) de la circunferencia

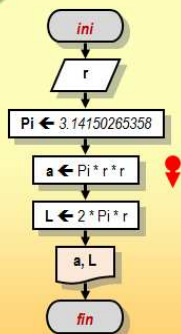
Fórmulas: $a = \pi r^2$ $L = 2\pi r$ $\pi = 3.14159265358$

Lógica: X

Elaborado por Elzabete - Docentes UTFRSA

Ejemplo

2 Desarrollo del Algoritmo



Note que en este caso las líneas de flujo siguen una dirección lineal y un sentido de arriba a abajo

seudocódigo C#

```

{
  r = Leer (dato)
  Pi = 3.14150265358
  a = Pi * r * r
  L = 2 * Pi * r
  mostrarMsj ( a, L )
}
  
```

Ejemplo

3 Codificación

Se verá próximamente

4 Validación y Prueba de escritorio

■ Se supondrá un valor de **radio = 5**, por lo que se tendrá:

Área = $3.1415926 * 5^2 = 3.1415926 * 25 = 78.539816$
 Long = $2 * 3.1415926 * 5 = 31.415926$

■ Se supondrá un valor de **radio = 10**, por lo que se tendrá:

Área = $3.1415926 * 10^2 = 3.1415926 * 100 = 314.15926$
 Long = $2 * 3.1415926 * 10 = 62.831852$

Una vez programado en un lenguaje se comprobará en la PC los resultados para los valores de radio dados.

Intercambio de Variables

Desde el punto de vista de **programación** será de la siguiente manera:

Dadas las variables numéricas enteras $v1=15$ (por ejemplo) y la variable $v2=36$ se quiere intercambiar sus valores por lo que se necesitará una variable auxiliar $v3$ para ello de forma que:

$v3 \leftarrow v1$ el contenido de $v1$ va a $v3$, ahora $v3$ contiene el valor 15

$v1 \leftarrow v2$ el contenido de $v2$ va a $v1$, ahora $v1$ contiene el valor 36

$v2 \leftarrow v3$ el contenido de $v3$ va a $v2$, ahora $v2$ contiene el valor 15

Estructuras Condicionales

■ Dependien del comportamiento de una determinada condición.

■ Los tipos de condicionales son:



exp = variable o expresión booleana (implica un resultado V o F).
 para el caso de **salida simple** si $exp = V$ entonces el flujo del algoritmo continúa por la rama indicada con **si**, al igual que en el caso de la **salida doble** y en esta última si $exp = F$ entonces el flujo continúa por la rama indicada con **no**.

V_i = variable o expresión de tipo # entera, carácter o cadena
 $c1, c2, etc.$ indican el camino a seguir acorde al valor de V_i

Ejemplo de salida múltiple

Representar mediante un **DF**, que emplee una condición de salida múltiple, la problemática siguiente:

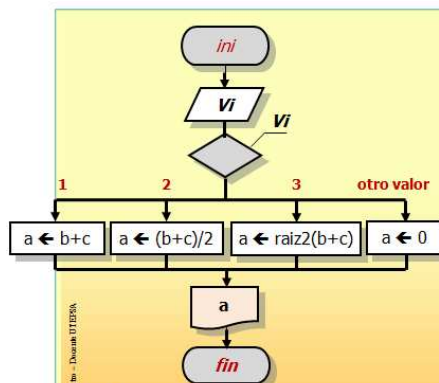
para $3 < V_i < 1$ entonces $a = 0$

si $V_i = 1$ entonces $a = b + c$

si $V_i = 2$ entonces $a = \frac{b+c}{2}$

si $V_i = 3$ entonces $a = \sqrt{b+c}$

Ejemplo de salida múltiple



seudocódigo C#

```

{
  Vi = Leer (dato)
  tomar (Vi)
  {
    caso 1: a = b + c
    caso 2: a = (b + c) / 2
    caso 3: a = raiz2(b + c)
    otro valor:
      a = 0
  }
  mostrarMsj ( a )
}
  
```

Note que las **constantes** son de tipo numérico entero ya que V_i es de este tipo

Ejemplos E. Condicionales

Elaborar un **DF** para un programa que espera tres (3) datos de entrada (a, b, c) teclados por el usuario de los que se quiere obtener el **valor máximo** y el **promedio** de los números.

1 Análisis

Datos: 3 valores numéricos **a, b, c**

Resultados: Valor máximo **Vmax**, valor promedio **Vprom**

Fórmulas: $Vprom = (a+b+c) / 3$

Lógica: si $a > b$ y $a > c$ entonces $Vmax = a$

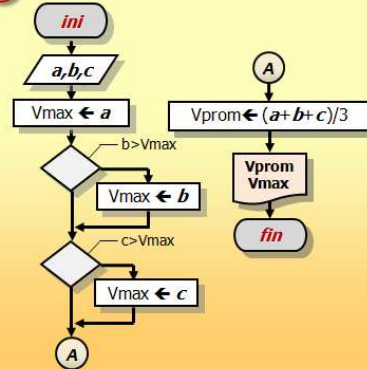
si $b > a$ y $b > c$ entonces $Vmax = b$

si $c > a$ y $c > b$ entonces $Vmax = c$

Elaborado por Flaminio - Docente UTEDSA

Ejemplo - 1

2 DF



seudocódigo C#

```
{
a = Leer (dato)
b = Leer (dato)
c = Leer (dato)
Vmax = a
si (b > Vmax) Vmax = b
si (c > Vmax) Vmax = c
Vprom = (a + b + c) / 3
mostrarMsj ( a )
}
```

Elaborado por Flaminio - Docente UTEDSA

Ejemplo

3 Codificación

Se verá en el próximo tema

4 Validación y Prueba de escritorio

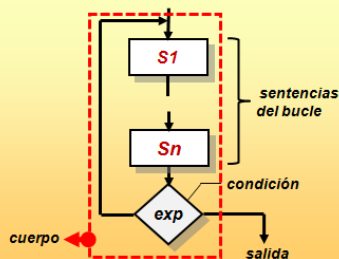
a	b	c	Máximo	Promedio
5	6	7	7	6
4	0	4	4	2.666....
0	-16	-2	0	-6

Una vez programado en un lenguaje se comprobará en la **PC** los resultados para los valores de números dados.

Elaborado por Flaminio - Docente UTEDSA

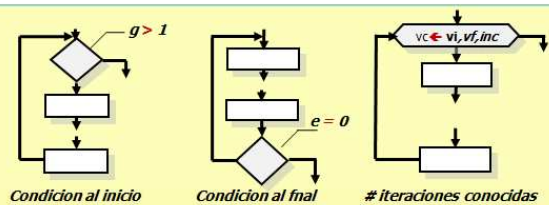
Estructuras Repetitivas – Bucles

- Un **bucle** es un **segmento** del programa que se **repite** un número determinado de veces **mientras se cumpla una condición determinada**.
- Las **partes** que conforman un **bucle** son:



Elaborado por Flaminio - Docente UTEDSA

Tipos de Bucles



- En caso de un Bucle con condición al inicio éste podría **NO** ejecutarse **NUNCA**, mientras que con condición al final éste se ejecuta **SIEMPRE** al menos **UNA VEZ**
- vc** representa la variable de control de las iteraciones (la cual se incrementa o disminuye en el valor dado en **inc** en cada iteración), mientras que **vi** y **vf** representan el rango o número de iteraciones a realizar

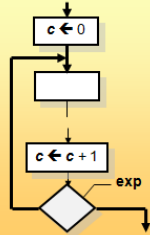
Elaborado por Flaminio - Docente UTEDSA

Contadores y Acumuladores

Un **contador** es una variable cuyo valor se **incrementa** o **disminuye** en una cantidad **constante** en cada iteración.

Características:

- Deben ser del tipo entero (positivos o negativos)
- Deben ser inicializados en un valor



Contadores y Acumuladores

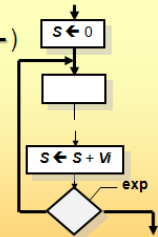
Un **acumulador** es una variable cuya misión es **almacenar cantidades variables** resultantes de una **sumatoria**.

Características:

- Pueden ser del tipo entero o real (+ o -)
- Deben ser inicializados en un valor

Notas:

Nótese que la inicialización del acumulador está fuera del cuerpo del Bucle. ($S \leftarrow 0$)
Vea además como el acumulador S va incrementando a su valor los distintos valores que toma V_i .



Ejemplo de Bucles

Calcular el **promedio** de una serie de números dados por el usuario desde el teclado. El **valor cero (0)** indica el **fin de la entrada de los datos**.

1 Análisis

Datos: valores numéricos $V_1, V_2, \dots, V_i, \dots, V_n$

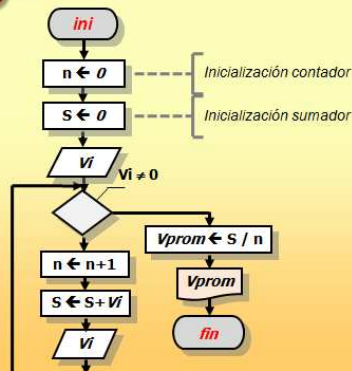
Resultados: valor **promedio** (V_{prom}) de los #s entrados

$$\text{Fórmulas: } V_{prom} = \frac{V_1 + V_2 + \dots + V_n}{n} = \frac{\sum_{i=1}^n V_i}{n} = \frac{S}{n}$$

Lógica: Empleo de un **acumulador S** para la Σ
Empleo de un **contador n** para contar los #s
Condición bucle $V_i \neq 0$ sigue la repetición

Ejemplo - 1

2 DF



seudocódigo C#

```
{
n = 0
S = 0
Vi = Leer (dato)
mientras (Vi ≠ 0)
{
n = n + 1
S = S + Vi
Vi = Leer (dato)
}
Vprom = S / n
mostrarMsj(Vprom)
}
```

Ejemplo

3 Codificación

Se verá en el próximo tema

4 Validación y Prueba de escritorio

Prueba de escritorio se comprobará mediante el siguiente juego de valores:

15, 8, 27, 102, 0 de donde
Promedio = $(15+8+27+102) / 4 = 152/4 = 38$

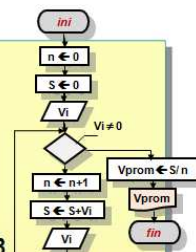
Validación del algoritmo (consiste en ir evaluando paso a paso las distintas variables acorde al flujo del diagrama)

Iteración 1 2 3 4 5 $V_{prom} = 152 / 4 = 38$

n = 0 1 2 3 4

S = 0 15 23 50 152

Vi = 15 8 27 102 0



PREGUNTAS de AUTOCONTROL:

- ¿Por qué son necesarios, en algunos casos, el uso de contadores en las estructuras repetitivas de los programas de computación?
- ¿Cuál es la diferencia entre dos bucles en uno de los cuales la condición se encuentra al inicio de éste y en otro en la que se encuentra al final del mismo?

EJERCICIOS PROPUESTOS:

1. Plantear los pasos lógico que usted sigue para ir desde su casa hasta su universidad.
2. Una persona solo sabe sumar dos números y dividir 2 números. Escribir los pasos que se deben seguir para que esta persona calcule el promedio entre los números 9, 4, 12.
3. Modificar el anterior- ejercicio para calcular el promedio de 3 números cualesquiera.
4. Calcular la velocidad promedio que emplea un corredor de maratón en su Carrera. Los datos del tiempo empleado, expresado en minutos, se introducen desde el teclado y el resultado se visualiza en pantalla en kilómetros por horas.
5. Convertir una temperatura expresada en grados Celsius a grados Fahrenheit. La formula de conversión es $F = 9/5 + 32$
6. Determinar el área y el volumen de un cilindro cuyas dimensiones se leen por teclado
7. Determinar si un número es múltiplo de 6 y 9 a la vez
8. Intercambiar dos variables a y b en caso de que a sea mayor a b
9. Determinar el mayor de tres números
10. Imprimir el día de la semana en caso de que su inicial coincida con una tecla oprimida por el usuario. En caso de que no coincida imprimir un mensaje de error.
11. Calcular el salario de un empleado si se sabe que para los de turno diurno la tarifa en el caso de horas normales de trabajo es de 1.3 \$/hora y en el caso de horas extras es de 2.1 si el empleado es de turno diurno, la tarifa será horas normales es de 1.8 y para horas extras es de 2.5.
12. Determinar si un carácter teclado por el usuario es un dígito, una mayúscula, una minúscula o un carácter especial
13. Determinar el mayor entre 4 números.
14. Generar los 10 primeros números naturales
15. Generar los primeros N número naturales.
16. Sumar los N primeros números naturales
17. Sacar el promedio de los primeros N números naturales.
18. Generar los N números enteros antes del 100.
19. Generar los N últimos números enteros antes de un número A.
20. De un conjunto de N números enteros, sacar el promedio de los pares.
21. De un conjunto de N números, sumar sólo aquellos que llegan en forma ascendente.
22. Escoger el menor entre N números
23. Escoger el mayor y menor de un conjunto de N números.
24. Generar las siguientes series:
 - a.) Serie de Fibonacci- 1, 1, 2,3, 5, 8, 13, 21, 34, 55, 89
 - b.) 1, 3, 4, 7, 11, 18, 29,
 - c.) 1, 20, 3, 18, 5, 16, 7, 14, 9, 12, 11, 10, 13, 8, 15, 6, 17, 4, 19, 21, 0
 - d.) 1, 2, 4, 8, 16, 32, 64, -128
 - e.) 1, 2, 3, 4, 5, 5, 4, 3, 2, 1, 1, 2, 3, 4, 4, 3, 2, 1, 1, 2, 3, 3, 2, 1, 1, 2, 2, 1, 1, 1
 - f.) -1, 1, -2, 3, -5, 8, -13, 21, -34, 55
 - g.) 1, 2, 6, 24, 120, 720n!
 - h.) -1, 1, -2, 2, -3, 4, -4, 8, -5, 16, -6, 32
25. Teniendo un número natural cualquiera, calcular la cantidad de dígitos que contiene.
26. Determinar el factorial de un número N

SISTEMA DE EVALUACIÓN

EVALUACIONES

Evaluaciones escritas

Evaluación	Descripción y temas a evaluar	Puntos	Fecha
Primer Parcial	<ul style="list-style-type: none"> Se evaluarán los conceptos teóricos y prácticos adquiridos en clase de los Sistemas Numéricos Posicionales SNP, tanto en las operaciones de conversión como en la aritméticas de Suma y Resta 	15	Clase 07
Trabajos Prácticos	<ul style="list-style-type: none"> Se plantean cuatro trabajos prácticos de: Conversiones de SNP Operaciones Aritméticas de SNP Evaluación y Representación de expresiones y cálculo de operaciones de Asignación. Planteamiento de los modelos de Algoritmos: Diagramas de Flujo y pseudocódigos orientado al lenguaje C#, de problemas sencillos y de mediana complejidad. 	15	
Segundo Parcial	<ul style="list-style-type: none"> Se evaluarán los conceptos teóricos y prácticos adquiridos en clase de la construcción de Diagramas de Flujo, así como pseudocódigos orientado al lenguaje C#, de problemas sencillos y de mediana complejidad 	20	Clase 15
Evaluación Final • Examen Final	<ul style="list-style-type: none"> Prueba integradora de conocimientos (en el examen final se tomara todo el contenido avanzado en la materia). Se evaluarán los conceptos y habilidades adquiridas en la construcción de algoritmos y pseudocódigo (C#) de problemas sencillos y de mediana complejidad. 	20	Clase 19
• Caso de Estudio.	<ul style="list-style-type: none"> Discusión de los CASOS de ESTUDIO CEs Se discute por los grupos de estudiante uno de los Casos de Estudio propuestos (selección al azar). Los CEs se suben al portal de UTEPSA en la segunda semana del módulo. 	30	Clase 20

Transversalización del idioma Inglés

Se sugiere al docente orientar un trabajo de clases de buscar el significado de palabras y estructuras de lenguajes **C#.NET**, como pudieran ser por ejemplo: referente a los tipos de datos, terminología de programación, sintaxis empleada en **C#.NET**, etc. El valor de los trabajos prácticos debe ser en total de **15 puntos**

MATERIAL COMPLEMENTARIO

Trabajo de Clases 1: Introducción a la Programación Docente: José Castro Medina **Ejercicios de Conversiones entre SNP**

I- Convertir de Binario a Octal y Hexadecimal los siguientes valores:

1. 010011101110_2
2. 111011101110_2
3. 1011011101111_2
4. 111101.110110100_2
5. 110011001001_2
6. 10011101111_2
7. 1010111100_2

II- Convertir a Binario los siguientes valores:

8. 257_8
9. 3521_{16}
10. 475_{16}
11. 2720_8
12. 3630_{16}
13. 1717_8
14. AE_{16}

III- Realizar las conversiones de Octal a Hexadecimal y viceversa siguientes:

15. 1234_8
16. $C0DE_{16}$
17. 567_8
18. BCE_{16}
19. $14AC_{16}$

IV- Realizar las conversiones siguientes

20. 1111_{10} a Octal y Hex
21. 1001_{10} a Octal y Hex
22. ABC_{16} a Bin y Octal
23. BAA_{16} a Bin y Octal
24. 5426_8 a Hex
25. 3702_8 a Hex
26. $F1A.3_{16}$ a Decimal
27. 100111011_2 a Decimal
28. $19B9_{16}$ a Octal
29. 3702_{16} a Decimal
30. 3005_{10} a Binario

Trabajo de Clases 2: Introducción a la Programación Docente: José Castro Medina

Ejercicios de Operaciones Aritméticas de SNP

Realice las operaciones aritméticas que se indican a continuación:

- 1- $CBA_{16} - ABC_{16}$
- 2- $2077_8 + 1234_8$
- 3- $11110000_2 + 10011111_2$
- 4- $505_8 + 247_8 - 141_8$
- 5- $20A4_{16} - BA_{16} + CC_{16}$
- 6- $10031_8 - 324_8$
- 7- $11DA_{16} + CA_{16}$
- 8- $101111010_2 - 10011111_2$
- 9- $180CB_{16} - B3BD_{16}$
- 10- $B16_{16} + 542_{16}$
- 11- $12A4_{16} + AA_{16}$
- 12- $7102_8 + 10031_8$
- 13- $CD0E_{16} - CA_{16} - A1_{16}$
- 14- $7505_8 + 247_8 - 324_8$
- 15- $AB_{16} + BC_{16} + CD_{16} + DE_{16}$
- 16- $11110000_2 + 10011111_2 - 1111010_2$
- 17- $AA_{16} + BB_{16} - CC_{16}$
- 18- $7562_8 - 247_8$
- 19- $12_8 + 23_8 + 34_8 + 45_8$
- 20- $FF_{16} - A1_{16} + EE_{16}$
- 21- $245_{16} + 61B_{16}$
- 22- $4A21_{16} - BA_{16} + F_{16}$
- 23- $10031_8 - 2017_8 + 11_8$
- 24- $E0DC_{16} + BD_{16} + 2B_{16}$
- 25- $5057_8 + 463_8 - 435_8$
- 26- $BAC_{16} + BA_{16} + AC_{16}$
- 27- $11010010_2 + 10011111_2$
- 28- $77_{16} + 88_{16}$
- 29- $2657_8 - 742_8$
- 30- $(21_8 + 32_8) - (34_8 + 45_8)$

Trabajo de Clases 3: Introducción a la Programación Docente: José Castro Medina

Ejercicios de Representación de la Información

1.- Escribir las expresiones algebraicas siguientes como expresiones algorítmicas:

a) $\sqrt{b^2 - 4ac}$

b) $\frac{4}{3}\pi r^2$

c) $\frac{x+y}{x} - \frac{3x}{5}$

d) $(x_2 - x_1) + (y_2 - y_1)$

e) $G = 3(a + b)^2$

f) $H = \frac{\sqrt{a+b}}{(x + \frac{y}{z})^3}$

g) $R1 = a - (cb^3 + 1)$

h) $R2 = x + \frac{ab+c}{d+e} - y$

i) $R3 = ab\sqrt{\frac{\sqrt{x-y}}{x+y}}$

j) $R4 = \frac{x - \sqrt{y}}{\sqrt{x+y}} + z$

k) $R5 = \frac{\frac{5}{ab}}{c}$

2.- Encontrar el valor de las siguientes expresiones:

a) $7 / 2$

b) $7 \% 2$

c) $13 / 3.0$

d) $7 * 10 - 5 \% 3 * 4 + 9$

e) $(7 * (10 - 5) \% 3) * 4 + 9$

f) $-4 * 7 + \text{pot}(2, 3) / 4 - 5$

g) $(\text{pot}(4, 2) + \text{pot}(2, 2)) < (10 + 5.0 / 2)$

h) $(\text{pot}(5, 2) > \text{pot}(2, 5)) \text{ o } (\text{pot}(3, 2) > \text{pot}(2, 3))$

i) $7 \% (5 \% 3)$

j) $((12 + 3) / 2) / (8 * (5 + 1))$

k) $\text{cuadrado}(\text{raiz}(4))$

l) $\text{trunc}(81.5) + \text{redondeo}(81.5)$

m) $!(\text{raiz}(25) == 5 + 2) \text{ y } (\text{pot}(0, 5) != \text{pot}(5, 0))$

n) $(\text{pot}(8, 2) - \text{pot}(2, 6)) < (5 - 26 / 5)$

o) $(1.0 * 3.0) >= \text{raiz}(9)$

p) $\text{cuadrado}(4) >= \text{abs}(16.49)$

q) $(24 \% 5) / 2.0 == 2$

3.- Encontrar el valor de las variables z en las expresiones siguientes:

a) $q \leftarrow 5$

$e \leftarrow 30$

$r \leftarrow 9$

$z \leftarrow (e - r \% 7) * q - r$

b) $w \leftarrow 9$

$x \leftarrow 2 + w \% 10$

$y \leftarrow w / 3$

$z \leftarrow x + w / y$

c) $w \leftarrow 6$

$x \leftarrow w * 10 - 6$

$y \leftarrow 7 + x / 6$

$z \leftarrow (w + x) / y$

d) $w \leftarrow 2$

$x \leftarrow (2 + w) \% 5$

$y \leftarrow w / 3.0$

$z \leftarrow x * w / y$

e) $q \leftarrow 3$

$e \leftarrow 7 * q + 1$

$r \leftarrow 8$

$z \leftarrow ((e + r) \% q - r) * 2$

f) $q \leftarrow 5$

$e \leftarrow 30$

$r \leftarrow 9$

$z \leftarrow ((e - r) \% 7) * (q - r)$

g) $w \leftarrow 9$

$x \leftarrow 2 + w / 10$

$y \leftarrow w / 3.0$

$z \leftarrow (x + w) / y$

h) $w \leftarrow 3$

$x \leftarrow (2 + w) \% 2$

$y \leftarrow w / 3$

$z \leftarrow x * w / y$

i) $q \leftarrow 2$

$e \leftarrow 7 * q + 1$

$r \leftarrow 8$

$z \leftarrow ((e + r) \% q - r) * 3$

Trabajo de Clases 4: Introducción a la Programación Docente: José Castro Medina

Ejercicios de Desarrollo de Algoritmos

(Los algoritmos deben ser planteados como **Diagramas de Flujo** y **Seudocódigo en C#**)

1. Cree un algoritmo para la conversión de temperaturas de forma que el usuario ingrese un valor de temperatura en grados centígrados (gC) y el algoritmo muestre el correspondiente en grados Fahrenheit (gF). Fórmula: $gC = \frac{5}{9}(gF - 32)$
2. Dado un conjunto de **n** números enteros dados por el usuario por el teclado, determinar el promedio de los números pares.
3. Dados **n** números enteros desde teclado, determinar la cantidad de positivos, negativos y múltiplos de 6.
4. De una colección de **n** exámenes con notas de 0 a 20 puntos, se desea calcular cuantos alumnos poseen notas de sobresaliente (15 a 20), cuantos con nota satisfactoria (10 a 14) y cuantos reprobados (0 a 9). Mostrar en pantalla los resultados obtenidos.
5. Para un concurso de belleza de la Universidad se necesita conocer la cantidad de alumnas que son mayores de **17 años** y menores de **31 años**, por lo que para la entrada de datos de los **n** alumnos de la Universidad, es necesario determinar, además de la edad, el sexo del alumno a través del **dato tipo carácter s** (s="F" para las alumnas y s="M" para los alumnos) y seleccionar aquellas alumnas que cumplen con el requisito de la edad. El resultado debe mostrarse en pantalla.
6. Cree una aplicación denominada **Sumatoria** que calcule la suma de los números de un determinado rango de valores (datos **n1** y **n2**) entrados por el usuario desde teclado y que se obtenga el resultado de dicha suma.
Ejemplo si **n1=10** y **n2=20** → **res=10+11+12+...+20=165**
7. Un comerciante ha realizado **n** ventas en un día y necesita conocer cuantas de éstas tienen un valor mayor a **Bs 500**, cuantas están entre **200 y 500** y cuantas por debajo de **Bs 200**. Diseñe un algoritmo donde se resuelva el problema planteado.
8. Calcular el salario de los **n** empleados de una fábrica, considerando que para los obreros del turno diurno la tarifa para el caso de **horas normales** (Hdianormal) es de **\$1.3 la hora**, mientras que para las **horas extras** (Hdiaextra) es de **\$2.1 la hora**, mientras que para las **horas nocturnas** (Hnochenormal) la tarifa es de **\$1.8 para horas normales** y de **\$2.5 para horas extras** (Hnocheextra). Diseñe un algoritmo (mediante un **DF**, plantee los pasos a seguir donde se resuelva el problema. Los datos entrados por el usuario desde teclado reflejan para cada trabajador la cantidad de horas trabajadas en cada horario
9. En una empresa se fabrican dos tipos de piezas metálicas; cónicas y cilíndricas. Para la confección de dichas piezas es necesario conocer el volumen de las mismas, luego para la producción de las **n** piezas de un día determine **la suma de volúmenes** de cada tipo de pieza sabiendo que las fórmulas de volumen para las mismas son:

Piezas Cilíndricas: $V = \pi r^2 H$;Piezas Cónicas $V = \frac{\pi}{3} r^2 H$

La entrada del usuario especifica los datos de cada tipo de pieza (tipo, radio y altura