

ALGORITMOS Y LENGUAJES DE PROGRAMACION

DISEÑO DE ALGORITMOS

**ESTRUCTURAS SECUENCIALES
ESTRUCTURAS NO SECUENCIALES
Sentencias selectivas**

DISEÑO DE ALGORITMOS

OBJETIVO

El objetivo principal de este Capítulo es capacitar al estudiante en los conceptos básicos de algoritmos, que le permitan obtener la destreza necesaria para diseñar sus propios algoritmos.

DISEÑO DE ALGORITMOS

- **CONCEPTOS BASICOS**
- **DESCRIPCION DE ALGORITMOS**
- **CONSTRUCCION DE ALGORITMOS**

CONCEPTOS BASICOS DE ALGORITMO

Un algoritmo es todo Programa, rutina ó método para resolver un problema; mediante una secuencia finita de acciones , cada cual con un significado concreto y cuya ejecución genera un tiempo finito.

La ventaja del algoritmo es que el programador se puede concentrar en la lógica del problema y en las estructuras de control del mismo. Dejando de lado las reglas de sintaxis del lenguaje de programación.

Algoritmo, es sinónimo de procedimiento computacional y es fundamental para la ciencia de la computación.

CONCEPTOS BASICOS DE ALGORITMO

PROPIEDADES:

- ***Posee entradas***

La entrada se toma como un conjunto especifico de valores que inicializan el algoritmo.

- ***Posee salidas***

Todo algoritmo posee una ó más salidas; la salida es la transformación de la entrada.

- ***Efectivo***

Un algoritmo es efectivo cuando las operaciones se efectúan de un modo exacto y en un tiempo finito aun usando métodos manuales.

DESCRIPCIÓN DE ALGORITMOS

Para describir un algoritmo se debe utilizar algún método que permita independizar dicho algoritmo del lenguaje de programación.

- **Diagramas de flujo**
- **PseudoCódigo**

DESCRIPCIÓN DE ALGORITMOS

Diagramas de flujo

Es una representación detallada en forma gráfica (mediante símbolos) de los pasos que se realizan para la solución de un problema unidos por flechas, denominadas líneas de flujo, que indican la secuencia de ejecución. (ya vimos en el tema anterior)

PSEUDOCÓDIGO

Es la descripción de un algoritmo mediante una secuencia lógica de acciones, que en su conjunto forman la solución del problema, utilizando generalmente palabras reservadas para representar las acciones.

PSEUDOCÓDIGO

La sencillez del lenguaje PseudoCódigo lo hacen ideal para la enseñanza de la programación.

Permitiendo el uso de valores numéricos, números decimales, lógicos, caracteres y arreglos.

Provee también funciones de entrada/salida y Funciones matemáticas.

PSEUDOCÓDIGO

Estructura de un Programa

Programa <nombre>

<secuencia de instrucciones>

FinPrograma

Donde una secuencia de instrucciones es una lista de una o más instrucciones, cada una terminada en punto y coma. <instrucción>; <instrucción>; ... <instrucción>;

PSEUDOCÓDIGO

Las palabras y símbolos que pertenecen al pseudocódigo se escriben tal como deben ir, por ejemplo: **Programa**; a diferencia de los nombres de variables y otros símbolos y expresiones que pueden variar.

PSEUDOCÓDIGO

Se pueden introducir comentarios luego de una instrucción, o en líneas separadas, mediante el uso de: // , #, /* */).

PSEUDOCÓDIGO

IDENTIFICADORES

Los *Identificadores* son palabras definidas por el programador para identificar variables.

Identificador, o nombre de variable, debe constar sólo de letras y números, comenzando siempre por una letra.

PSEUDOCÓDIGO

TIPO DE DATOS

a) Tipos Simples:

Numérica, números enteros y decimales,
se usa el punto para separar los
decimales.

Carácter caracteres o cadena de caracteres
se escribe entre comillas (“ “) ó (‘ ‘)

Lógicas toma dos valores: Verdadero ó Falso

b) Datos Compuestos. Estructura de Datos

PSEUDOCÓDIGO

PALABRAS RESERVADAS

Leer , Escribir , Dimension

Si-Entonces-Sino, Segun

Mientras Hacer, Repetir Hasta Que, Para

PSEUDOCÓDIGO

ESTRUCTURAS SECUENCIALES

Las estructuras secuenciales están formadas por las siguientes instrucciones:

- ❖ **Leer** **Entrada de datos**
- ❖ **Escribir** **Salida de Resultados**
- ❖ **Asignación** **Almacena un valor en una variable**

PSEUDOCÓDIGO

ESTRUCTURA SECUENCIAL

La entrada de datos

Definida por la Instrucción **Leer** que permite el ingreso de datos al algoritmo desde el teclado.

Leer <variable1>,<variable2>,..., <variableN>;

Leer Numero;

Leer n1, n2, n3;

PSEUDOCÓDIGO

ESTRUCTURAS SECUENCIALES

La salida de Resultados

Definida por la Instrucción **Escribir** y permite mostrar resultados en la pantalla.

Escribir <expr1> , <expr2> , ... , <exprN> ;

Escribir "Ingresa el nombre:";

Escribir "Resultado:" , $x*2$;

PSEUDOCÓDIGO

ESTRUCTURA SECUENCIAL

Asignación

La Instrucción de Asignación permite almacenar un valor en una variable.

<variable> ← <expresión> ;

PSEUDOCÓDIGO

Para ejecutarse la **asignación**, primero se evalúa la expresión de la derecha y luego se asigna el resultado a la variable de la izquierda. El tipo de la variable y el de la expresión deben coincidir.

$n \leftarrow 1;$ // asigna el número 1 a n

$\text{suma} \leftarrow \text{suma} + 1;$ // incrementa el valor de
suma en 1

PSEUDOCÓDIGO

Sentencia de Asignación

Ejemplo de sentencia de asignación:

$a \leftarrow 5$

$\text{suma} \leftarrow (2 + 3) * X + T + Z$

La jerarquía de los operadores matemáticos es igual a la del álgebra, aunque puede alterarse con el uso de paréntesis.

PSEUDOCÓDIGO

Sentencia de Asignación

Como ejemplos de sentencias de asignación podemos mencionar:

Velocidad \leftarrow (distancia/(minuto*60+segundo+c/100))

Numero \leftarrow u + 100 + d+10 + c

AT \leftarrow (s+(s-1) + (s-2) + (s-3))^2

PSEUDOCÓDIGO

Expresiones

Las expresiones son combinaciones (asociaciones) de datos con **operadores**.

$a \leftarrow 3$

$b \leftarrow 5$

$a + b - 7$

Cada expresión **toma un valor** que se determina al evaluar la expresión, el valor puede ser de tipo: numérico, lógico y carácter.

PSEUDOCÓDIGO

Expresiones

De acuerdo a los tipos de datos que se manipulan en las expresiones pueden ser:

Expresiones aritméticas (devuelve un número)

Expresiones lógicas (devuelve un valor lógico)

Expresiones de carácter (devuelve un carácter)

suma \leftarrow **a + b - 3** (expresión aritmética)

PSEUDOCÓDIGO

Tipos de Operadores

Un operador indica el tipo de operación a realizar sobre los datos.

Los operadores se clasifican en:

Operadores Aritméticos

Operadores relacionales

Operadores lógicos.

PSEUDOCÓDIGO

Expresiones Aritméticas

| <i>Operador</i> | <i>Acción</i> | <i>Ejemplo</i> | <i>Resultado</i> |
|-----------------|-----------------------|----------------|------------------|
| - | <i>Resta</i> | $5 - 2$ | 3 |
| + | <i>Suma</i> | $2 + 3$ | 5 |
| * | <i>Multiplicación</i> | $4 * 5$ | 20 |
| / | <i>División</i> | $9 / 3$ | 3 |
| ^ | <i>Potencia</i> | $5 ^ 2$ | 25 |

E
siguientes operadores aritméticos:

PSEUDOCÓDIGO

Expresiones Aritméticas

Los operadores aritméticos nos permiten, básicamente, hacer cualquier operación aritmética (suma, resta, multiplicación y división)

- $3 * 6/3$ resultado - 6
- $3 * (6/2)$ resultado - 9

Expresiones Lógicas

Las expresiones lógicas son asociaciones de datos con operadores relacionales y de su evaluación siempre se obtiene un valor de tipo lógico (verdadero o falso).

| <i>Operador Relacional</i> | <i>Acción</i> | <i>Ejemplo</i> |
|----------------------------|-------------------|----------------|
| > | Mayor que | 5 > 3 |
| >= | Mayor o igual que | a >= b |
| < | Menor que | 2 < 4 |
| <= | Menor o igual que | a <= b |
| = | Igual que | m = n |
| <> | Distinto que | 3 <> 2 |

PSEUDOCÓDIGO

Expresiones Lógicas

El operador conjunción (y) y el operador disyunción (o) siempre actúan sobre dos operandos, mientras que, el operador negación (no) sólo actúa sobre un operando.

| <i>Operadores Lógicos</i> | <i>Acción</i> | <i>Ejemplo</i> |
|-------------------------------|-------------------------|---------------------------------------|
| & | Conjunción (y) | (5>3) & (5=1) // falso |
| | Disyunción (o) | (7=7) (4>9) // verdadero. |
| NO | Negación (no) | NO (4=4) // falso |

Expresiones Lógicas

Los operadores lógicos producen un resultado booleano, y sus operandos son también valores lógicos.

| <i>Operadores Lógicos</i> | <i>Acción</i> | <i>Ejemplo</i> |
|----------------------------------|----------------------|-----------------------|
| Y | Conjunción | (5>2) Y (4>1) |
| O | Disyunción | (2=2) O (a=b) |
| NO | Negación | No (9>3) |

PSEUDOCÓDIGO

Expresiones Lógicas

Tabla de verdad del operador Y:

| <code><expresión_1></code> | <code><expresión_2></code> | <code><expresión_1> y <expresión_2></code> |
|----------------------------------|----------------------------------|--|
| verdadero | verdadero | verdadero |
| verdadero | falso | falso |
| falso | verdadero | falso |
| falso | falso | falso |

PSEUDOCÓDIGO

Expresiones Lógicas

Tabla de verdad del operador O:

| <code><expresión_1></code> | <code><expresión_2></code> | <code><expresión_1> o <expresión_2></code> |
|----------------------------------|----------------------------------|--|
| verdadero | verdadero | verdadero |
| verdadero | falso | verdadero |
| falso | verdadero | verdadero |
| falso | falso | falso |

PSEUDOCÓDIGO

Expresiones Lógicas

Tabla de verdad del operador negación (no):

| <code><expresión></code> | <code>no <expresión></code> |
|--------------------------------|-----------------------------------|
| verdadero | falso |
| falso | verdadero |

Evaluar :

no (9 > 5) se obtiene como resultado falso

PSEUDOCÓDIGO

Prioridad de los operadores

Prioridad de los Operador, aritméticos, relacionales y lógicos

()

Paréntesis

\wedge

Potencia

$*$, $/$, **NO**

Multiplicación, división y negación

$+$, $-$, **Y**

Signo más, signo menos y conjunción

$>$, $<$, $>=$, $<=$, $<>$, $=$, **O** ***Mayor que, menor que, mayor igual que, menor igual que, igual que, distinto que y disyunción***

Funciones Matemáticas

| FUNCIÓN | SIGNIFICADO |
|-----------------|---------------------------------|
| RC(X) | Raíz Cuadrada de X |
| ABS(X) | Valor Absoluto de X |
| LN(X) | Logaritmo Natural de X |
| EXP(X) | Función Exponencial de X |
| SEN(X) | Seno de X |
| COS(X) | Coseno de X |
| ATAN(X) | Arcotangente de X |
| TRUNC(X) | Parte entera de X |
| REDON(X) | Entero más cercano a X |

La función raíz cuadrada no debe recibir un argumento negativo. La función exponencial no debe recibir un argumento menor o igual a cero.

CONSTRUCCION DE ALGORITMOS

Programa Problema1

Leer a , b;

Escribir "a = ", a;

Escribir "b = ", b;

suma <- a + b;

resta <- a - b;

multiplica <- a*b;

dividir <- a/b;

Escribir "suma = " , suma;

Escribir "resta = " , resta;

Escribir "multiplica = " , multiplica;

Escribir "dividir = " , dividir;

FinPrograma

CONSTRUCCION DE ALGORITMOS

PROBLEMA N° 2

Leer el radio de un circulo y calcular e imprimir su superficie y la longitud de la circunferencia.

Programa Problema2

pi <- 3.1416;

Leer radio;

Superficie <- pi * radio² ;

Circunferencia <- 2 * pi * radio;

Escribir "Radio", radio;

Escribir "Superficie", superficie;

Escribir "Circunferencia", circunferencia;

FinPrograma

PSEUDOCÓDIGO

PROBLEMA N°3

2

Calcular las raíces reales de una ecuación cuadrática $ax^2 + bx + c = 0$

Programa problema3

Escribir "introduzca las variables a, b y c";

Leer a, b, c;

$x1 \leftarrow (-b + \text{RC}((b^2) - (4*a*c)) / (2*a));$

$x2 \leftarrow (-b - \text{RC}((b^2) - (4*a*c)) / (2*a));$

Escribir "las raices son: ", x1, x2;

FinPrograma

PSEUDOCÓDIGO

ESTRUCTURAS NO SECUENCIALES

SENTENCIAS SELECTIVAS:

Selección Simple Si-Entonces

Selección Doble Si-Entonces-Sino

Sentencia Múltiple Segun o Case

SENTENCIAS REPETITIVAS:

Mientras Hacer

Repetir _ Hasta_ Que

Para

PSEUDOCÓDIGO

SENTENCIAS DE SELECCION

Se utiliza cuando en el desarrollo de la solución de un problema debemos **tomar una decisión**, para seleccionar una alternativo a seguir.

La **toma de decisión** se basa en la evaluación de una o más **condiciones** que nos permitan **seleccionar** una alternativa de solución al problema planteado.

PSEUDOCÓDIGO

a) SELECCION SIMPLE : *Si-Entonces*

Consiste en evaluar una condición; si el resultado de dicha evaluación es **Verdad**, se ejecuta la **sentencia por verdad** y termina la estructura.

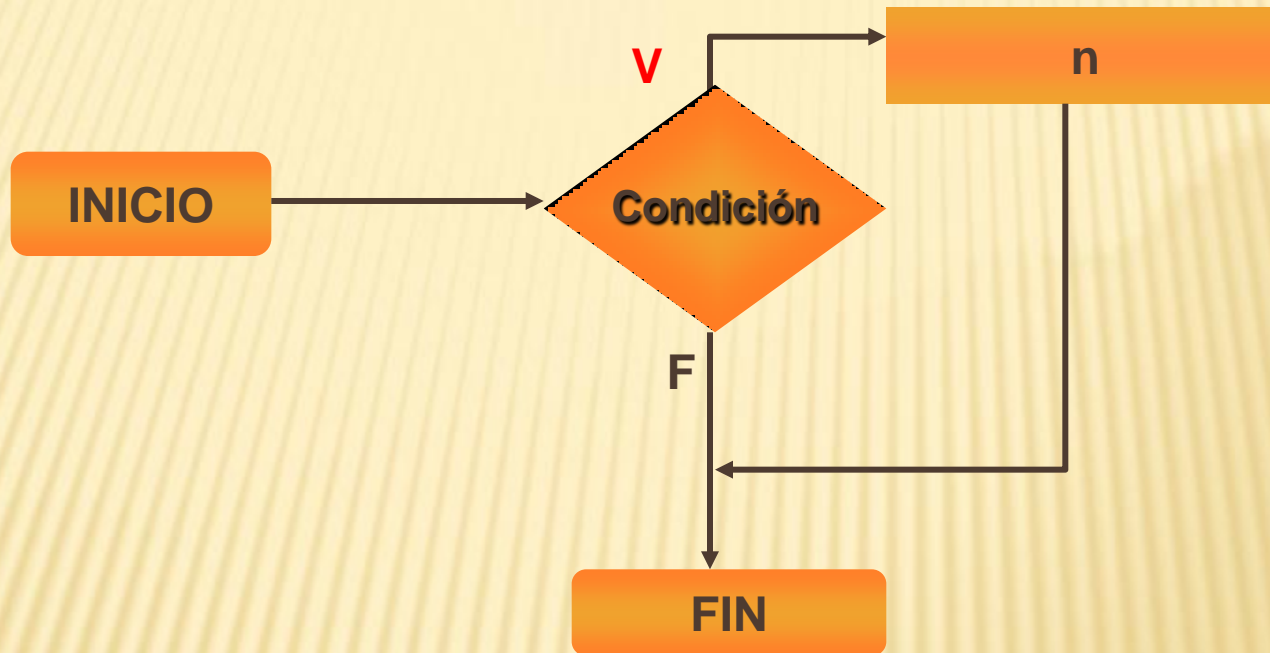
Si <condición> Entonces

Sentencia ejecutada si la condición es verdad

finSi

Si (nota >= 10) Entonces
Escribir ("ingreso")
fin Si

PSEUDOCÓDIGO



*La sentencia condicional o selectiva evalúa una **CONDICION** (expresión lógica)*

CONSTRUCCION DE ALGORITMOS

PROBLEMA N° 4

Escribir un algoritmo que lea los datos: a , b y calcule e imprimir la suma, la resta, la multiplicación y la división de ellos.

CONSTRUCCION DE ALGORITMOS

Programa Problema4

Leer a, b;

suma<- a+b;

resta<- a-b;

multiplica <- a*b;

Si b>0 **Entonces**

division <- a/b;

FinSi

Escribir "suma = ", suma;

Escribir "resta = ", resta;

Escribir "multiplica = ", multiplica;

Escribir "division = ", division;

FinPrograma

PSEUDOCÓDIGO

b) SELECCION DOBLE : ***Si-Entonces-Sino***

La sentencia de **selección doble** nos permite tomar decisiones entre dos alternativas luego de evaluar una condición (expresión lógica).

Si <condición>

Entonces

<instrucciones>

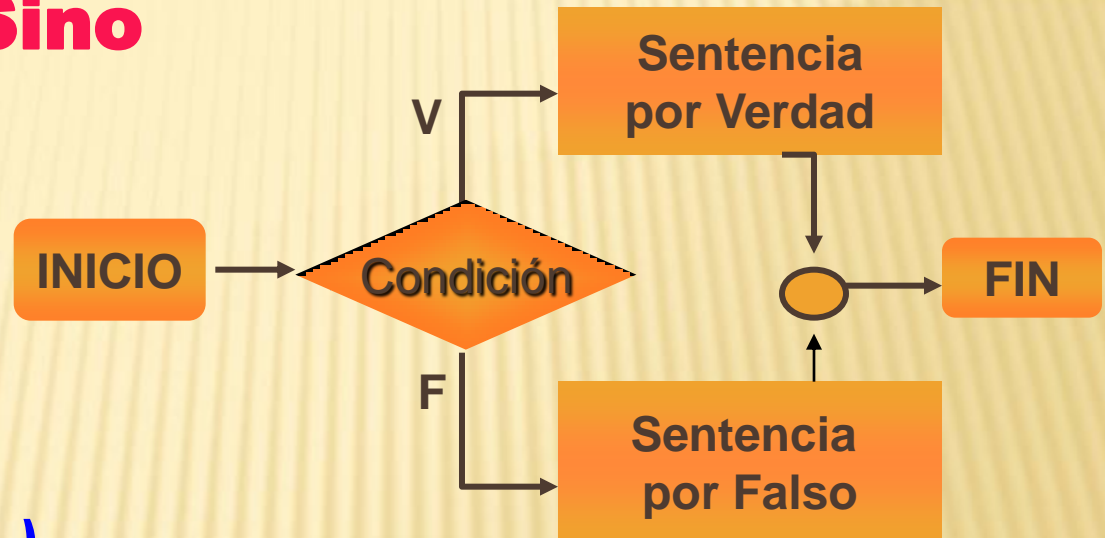
Sino

<instrucciones>

Finsi

PSEUDOCÓDIGO

Si - Entonces - Sino



Si (*nota* ≥ 10)

Entonces

Escribir (“ingreso”)

Sino

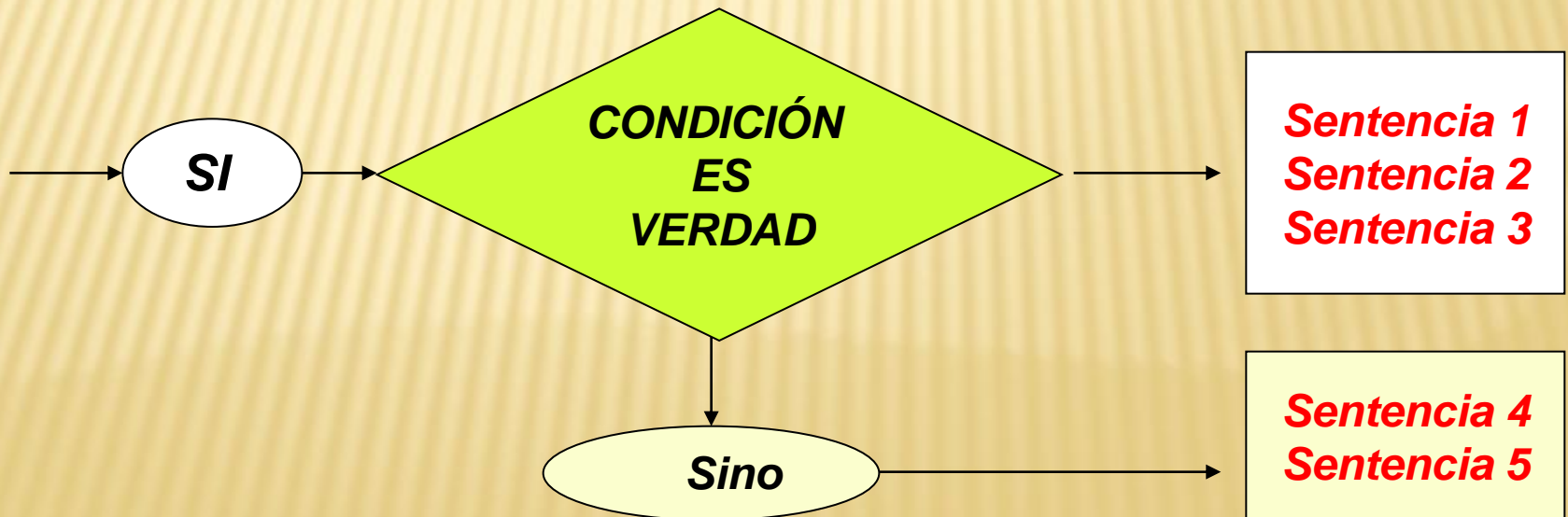
Escribir(“no-ingreso”)

Finsi

PSEUDOCÓDIGO

b) SENTENCIA DE SELECCIÓN DOBLE :

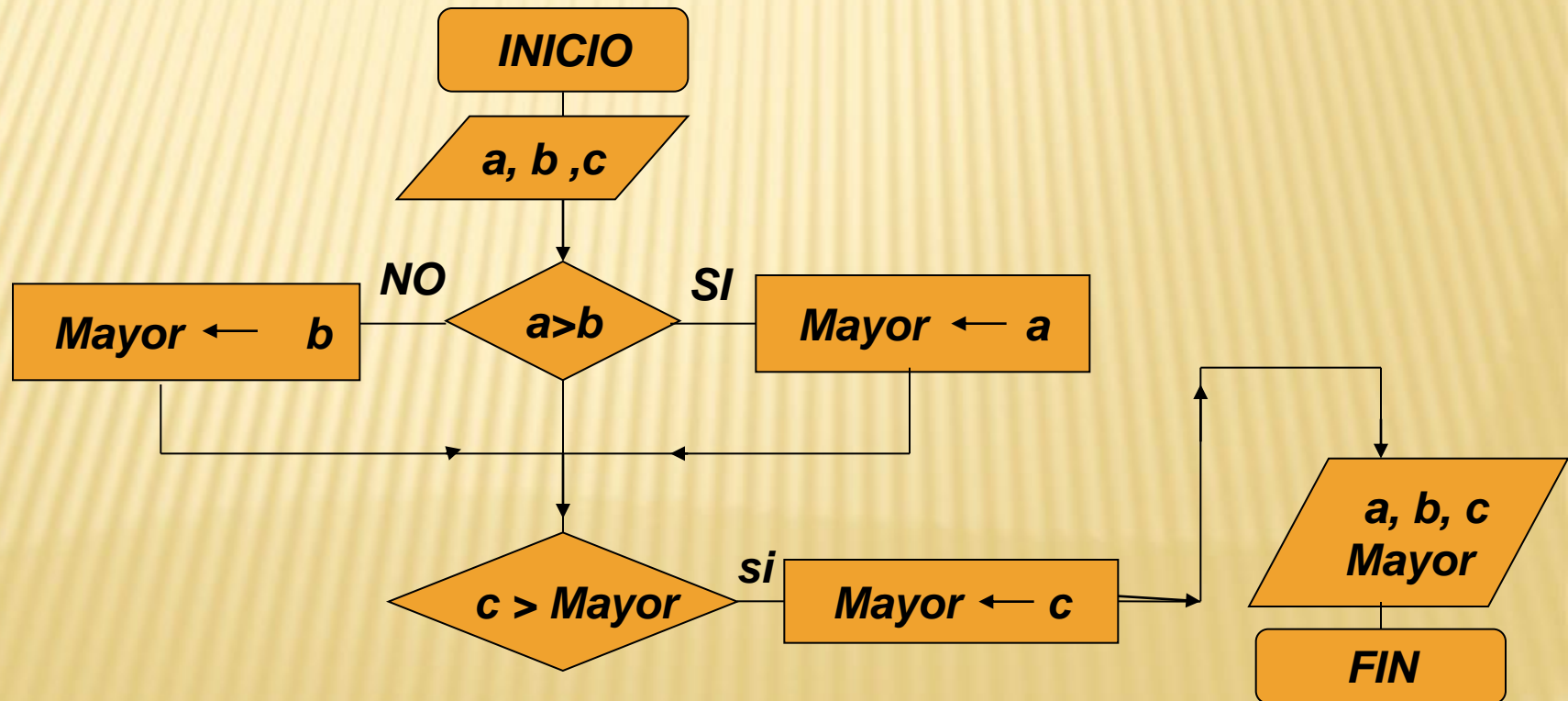
La sentencia de selección doble evalúa una condición e implica la selección de una de dos alternativas.



PSEUDOCÓDIGO

PROBLEMA N° 5

Escribir un algoritmo que lea tres números enteros positivos a , b y c e imprima el mayor de ellos.



CONSTRUCCION DE ALGORITMOS

Programa Problema5

Leer a, b, c;

Si $a > b$ Entonces mayor $\leftarrow a$;

Sino mayor $\leftarrow b$;

FinSi

Si $c > \text{mayor}$ Entonces mayor $\leftarrow c$;

Finsi

Escribir "a = ", a;

Escribir "b = ", b;

Escribir "c = ", c;

Escribir "mayor = ", mayor;

FinPrograma

PSEUDOCÓDIGO

Es posible utilizar la sentencia **Si** para diseñar estructuras de selección con más de dos alternativas.

Una estructura de selección de **n** alternativas puede ser construida utilizando una estructura **Si** .

Las estructuras **Si** interiores a otras estructuras **Si** se denominan anidadas o encajadas. Debe existir una correspondencia entre las palabras reservadas **Si** y **Finsi**.

PSEUDOCÓDIGO

SENTENCIA DE SELECCIÓN ANIDADAS

Si (condición1) Entonces <instruccion1>

Sino

Si (condición2) Entonces <instrucción2>

Sino

Si (condición3) Entonces
<instruccion3>

Finsi

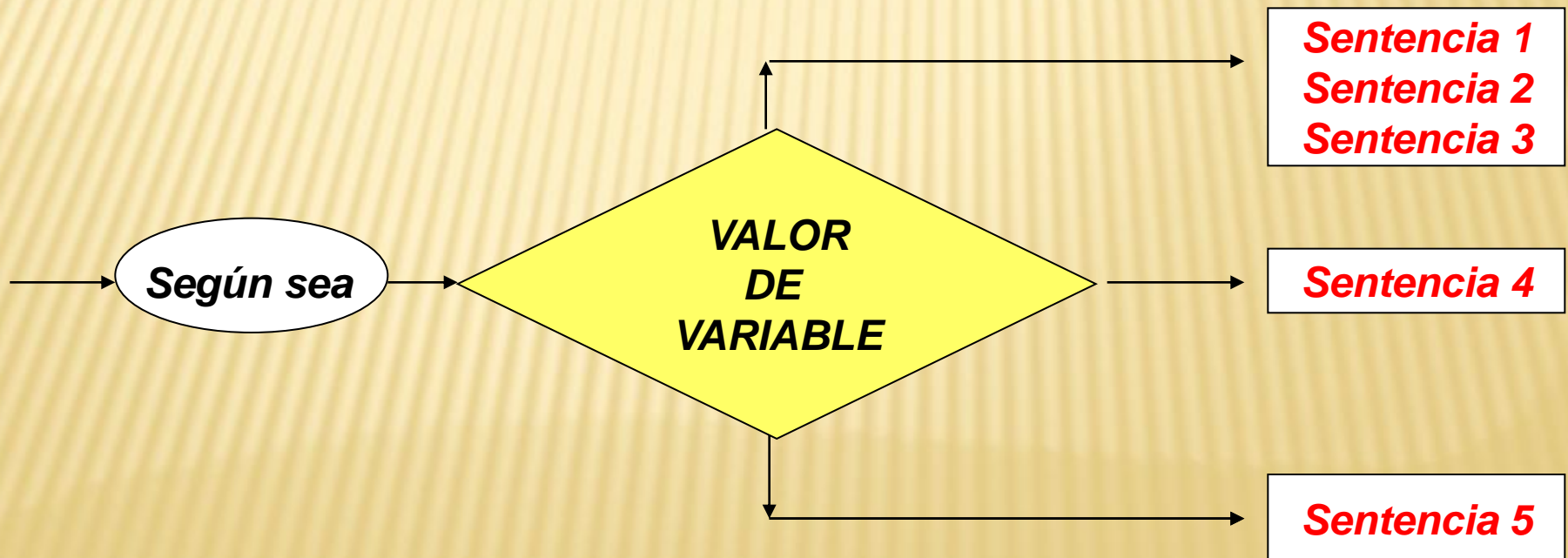
Finsi

Finsi

PSEUDOCÓDIGO

c) SENTENCIA DE SELECCIÓN MULTIPLE (según o case)

La sentencia de selección múltiple, se usa para seleccionar una de varias alternativas.



PSEUDOCÓDIGO

C) SELECCION MULTIPLE

SENTENCIA Segun



La selección Múltiple se utiliza cuando existen más de dos alternativas para la **selección** de una alternativa.

El selector puede ser una **variable** simple o una **expresión** simple denominada **expresión de control**.

PSEUDOCÓDIGO

C) SELECCION MULTIPLE : SENTENCIA Segun

Según/Case variable_numerica **Hacer**

opcion_1:

secuencia_de_acciones_1

opcion_2:

secuencia_de_acciones_2

opcion_3:

secuencia_de_acciones_3

De Otro Modo:

secuencia_de_acciones_dom

FinSegun/FinCase

CONSTRUCCION DE ALGORITMOS

Programa Problema6

leer n1, n2;

leer opcion;

segun opcion Hacer

1: suma <- n1 + n2;

Escribir "suma = ", suma;

2: resta <- n1 - n2;

Escribir "resta = ", resta;

3: multiplica <- n1*n2;

Escribir "multiplica = ", multiplica;

4: divide <- n1/n2;

Escribir "divide = ", divide;

5: potencia <- n1^n2;

Escribir "potencia = ", potencia;

De Otro Modo:

Escribir "Numero no Definido" ;

FinSegun

FinPrograma

CONSTRUCCION DE ALGORITMOS

PROBLEMA N° 7

Diseñar un algoritmo que lea un entero positivo n y realice las siguientes cálculos:

Si N termina en 0 se cambia por el doble

Si N termina en 3 se cambia por la décima parte

De otro modo mantiene su valor inicial

CONSTRUCCION DE ALGORITMOS

Programa Problema7

Leer n;

Segun n mod 10 **Hacer**

0: n <- n*2;

Escribir "Doble de N = ", n;

3: n <- n/10;

Escribir "Decima parte de N = ", n;

De Otro Modo:

n <- n;

Escribir "Numero N = ", n ;

FinSegun

FinPrograma