



# DISEÑO DE ALGORITMOS

## TEMA 1. ESTRUCTURAS DE CONTROL SECUENCIAL Y SELECTIVAS CON DIAGRAMAS DE FLUJO Y PSEUDOCÓDIGO

Presenta: Mtro. David Martínez Torres

Universidad Tecnológica de la Mixteca

Instituto de Computación

Oficina No. 37

[dtorres@gs.utm.mx](mailto:dtorres@gs.utm.mx)

# Contenido

1. Conceptos: algoritmo, tipos de datos, operadores, variables y constantes.
2. Estructura de control Secuencial.
3. Estructura de control Selectiva simple.
4. Estructura de control Selectiva doble.
5. Estructura de control Selectiva múltiple.

# 1. Conceptos: algoritmo

Un **algoritmo** es un conjunto de pasos, procedimientos o acciones que nos permiten alcanzar un resultado o resolver un problema [1]



Un **algoritmo** es un método para resolver un problema mediante una serie de pasos precisos, definidos y finitos [2, 3].

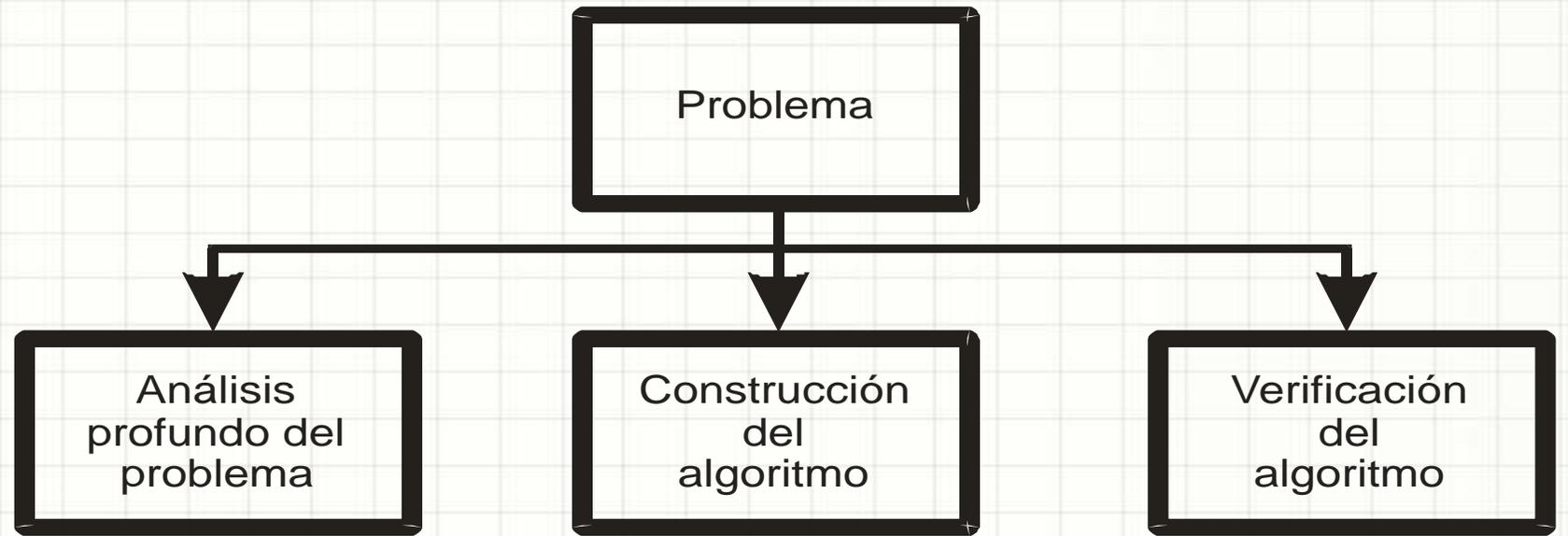
1. **Preciso:** cada paso debe indicar de manera precisa que se debe hacer.
2. **Definido:** Si se prueba con las mismas entradas debe producir los mismos resultados.
3. **Finito:** Debe tener un número determinado de pasos; debe producir un resultado en un tiempo finito.

# 1. Conceptos: algoritmo



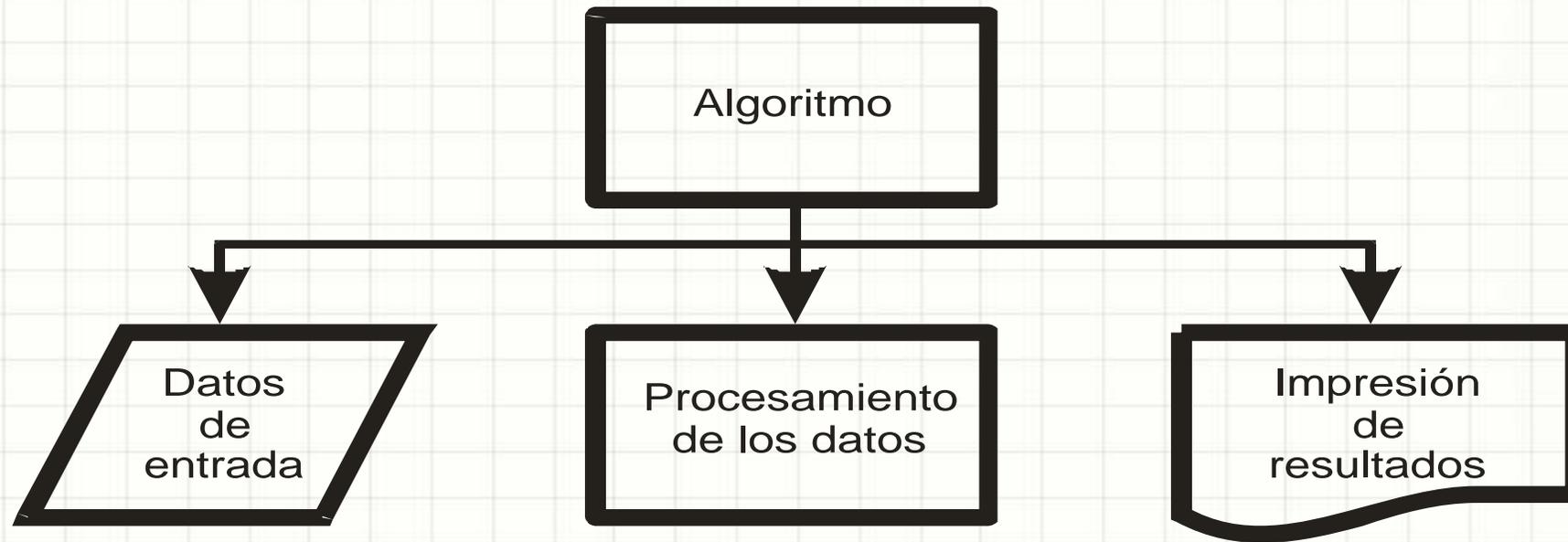
En la vida diaria aplicamos algoritmos de manera inconsciente: al vestirse, al abrir la puerta, al darse un baño, al ir a la escuela, al subir al automóvil, al cambiar la llanta de un automóvil, etc.

# 1. Conceptos: los algoritmos como parte de una metodología para la solución de problemas por computadora



Pasos para resolver un problema a través de un algoritmo, reeditado de [2]

# 1. Conceptos: Componentes de un algoritmo



Secciones de un algoritmo, reeditado de [2, Cairó Osvaldo]



**Problema 1.** Dado por teclado dos números enteros, calcular e imprimir la suma de ellos.

**Entrada:** introducir dos números, por ejemplo 4 y 8

variables num1, num2: entero

4		8
<b>num1</b>		<b>num2</b>

**Proceso:** Ejecutar la operación suma

variable suma: entero

12	=	4	+	8		4
<b>suma</b>		<b>num1</b>		<b>num2</b>		<u>+ 8</u>
						<b>12</b>

**Salida:** imprimir el valor de la suma que es igual a 12.

## Análisis y modelado de un problema

Este primer algoritmo presenta una secuencia de pasos coherentes y dependientes.

**Coherentes** debido que obtiene un valor de la forma correcta, como siempre se efectúa una suma.

**Dependiente** en que no se puede sumar dos números que no se conocen, y obtener el resultado de la suma sin sumar los números.

## Análisis y modelado de un problema



**Problema 2.** Dado por teclado un número entero, y a ese número le suma 10 e imprime el resultado.

**Entrada:** introducir un número, por ejemplo 20

variable num: entero

20

num

**Proceso:** Ejecutar la operación suma

variable suma: entero

20

30

20

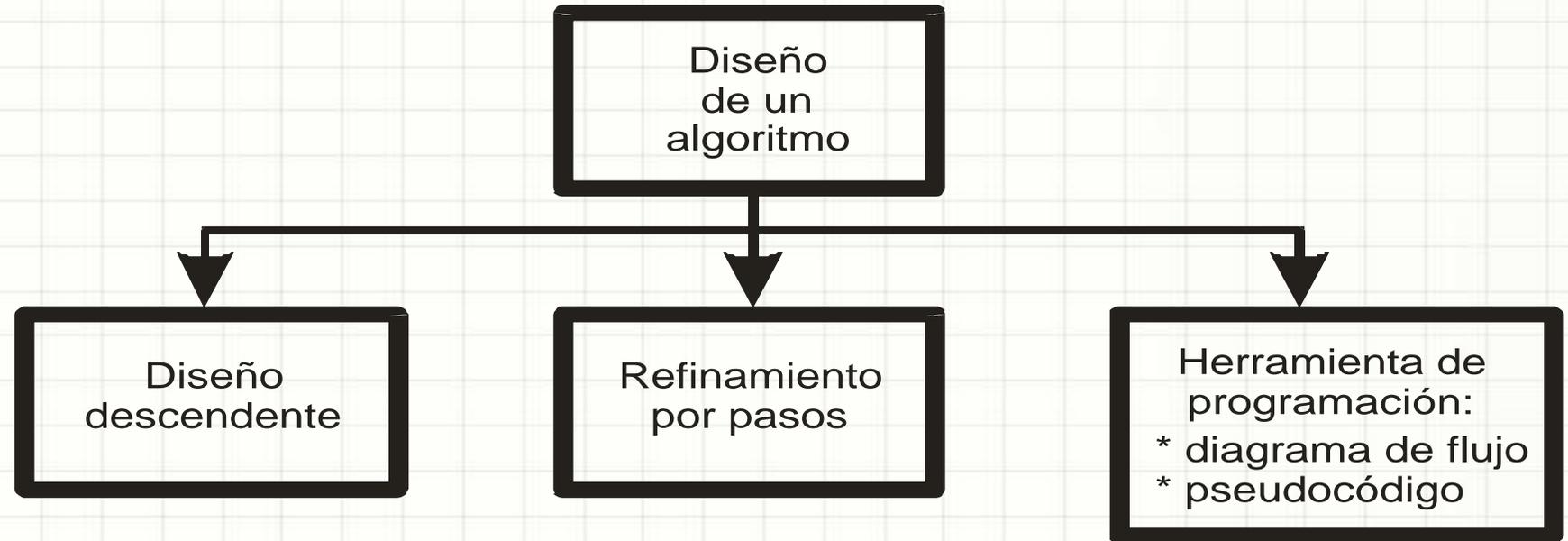
+ 10

suma = num + 10

30

**Salida:** imprimir el valor de la suma que es igual a 30.

# 1. Conceptos: Diseño del algoritmo



Secciones de un algoritmo, reeditado de [2]

**Problema 1.** Dado por teclado dos números enteros, calcular e imprimir la suma de ellos.

**Entrada:** introducir dos números, por ejemplo 4 y 8

variables num1, num2: entero

4	8
num1	num2

**Proceso:** Ejecutar la operación suma

variable suma: entero

12	4	8
suma	= num1	+ num2

**Salida:** imprimir el valor de la suma que es igual a 12.

## Diseño del algoritmo

### Algoritmo. Suma de dos números enteros

1. Inicio
2. Variables num1, num2, suma: entero
3. Imprimir Algoritmo que suma dos números enteros
4. Pedir al usuario que teclee dos números.
5. Leer los dos números en las variables num1 y num2.
6. Sumar num1 con num2 y poner el resultado en la variable suma.
7. Imprimir el resultado.
8. Fin

## Algoritmo. Suma de dos números enteros

1. Inicio
2. Variables num1,num2, suma: entero
2. Imprimir Algoritmo que suma dos números enteros
3. Pedir al usuario que teclee dos números.
4. Leer los dos números en las variables num1 y num2.
5. Sumar num1 con num2 y poner el resultado en la variable suma.
6. Imprimir el resultado.
7. Fin

## Verificación del algoritmo (Prueba de escritorio)



### Prueba de escritorio.

num1	num2	suma	salida
-	-	-	<b>Teclea 2 números: 4</b>
<b>4</b>	<b>8</b>	<b>12</b>	<b>8</b> <b>La suma es: 12</b>

# 1. Conceptos: Tipos de datos



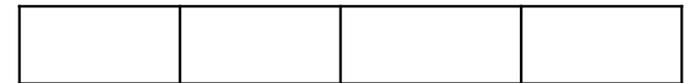
- 1. Tipos de datos.** Necesarios para representar los datos a procesar en un programa, y se clasifican en:
- a) **Simple:** enteros, reales, caracter, y booleanos.
  - b) **Estructurados:** arreglos, cadena de caracteres y registros.

identificador



Dato simple

identificador



Dato estructurado

# 1. Conceptos: Tipos de datos

## Ejemplos de datos:

**Enteros:** 128, 1500, -200, 8432, -14321, etc.

**Reales:** 6.4, 138.25, -32.453, 130.65, -8729.32, etc.

**Caracter:** letras del abecedario, dígitos, símbolos especiales(#,\$,^, \*,%,/,!,+,-, etc). Cada carácter debe estar cerrado entre apóstrofe o comillas sencillas: 'a', 'B', '\$', '9', '-'

**Cadena de caracteres:** Son encerrados con comillas dobles, ej. "abcd", "Pedro", "5326745", etc.

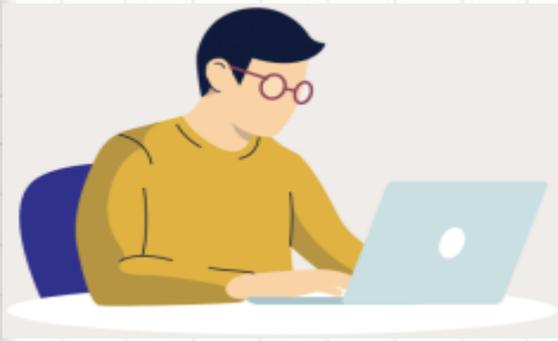


# 1. Conceptos: Definiciones de identificadores, constantes y variables

**Identificadores:** representan los nombres de variables, constantes y funciones donde almacenan los datos.

**Constantes:** son identificadores que su valor no cambia durante la ejecución de un programa. Ejemplo PI 3.1416, TAMANIO 10, NOMBRE "PESO"

**Variables:** son identificadores que su valor normalmente cambia durante la ejecución de un programa. Pueden ser de los tipos que se estudiaron en la diapositiva anterior.



# 1. Conceptos: Operaciones aritméticas

Operadores Aritméticos			
Operador aritmético	Operación	Ejemplo	Resultado
$\wedge$	potencia	$4^3$	64
*	Multiplicación	$8.25*7$	57.75
/	División	$15/4$	3.75
+	Suma	$125.78+62.50$	188.28
-	Resta	$65.30-32.33$	32.97
mod	Residuo	$15 \text{ mod } 2$	1

# 1. Conceptos: Jerarquía de los operadores aritméticos

Jerarquía de los Operadores Aritméticos		
Operador	Jerarquía	Operación
$\wedge$	Mayor	Potencia
$*, /, \text{mod}$	$\downarrow$	Multiplicación, División, Módulo
$+, -$	Menor	Suma, Resta

# 1. Conceptos: Operadores relacionales

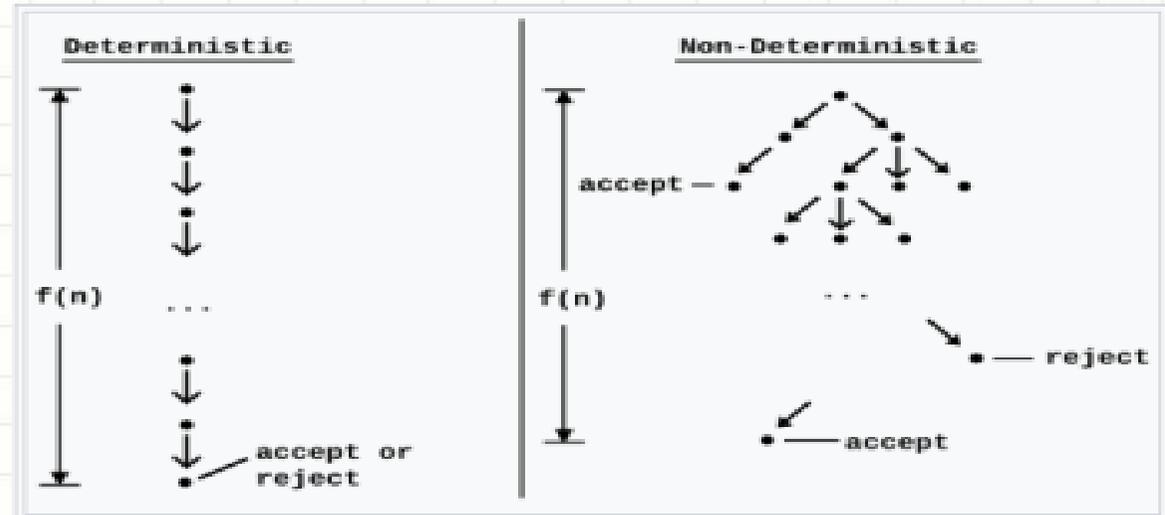
Operadores Relacionales			
Operador	Operación	Ejemplo	Resultado
=	Igual que	$10 = 11$	FALSO
$\neq$	Diferente a	$5 \neq 4$	VERDADERO
<	Menor que	$7 < 15$	VERDADERO
>	Mayor que	$22 > 11$	VERDADERO
$\leq$	Menor o igual que	$15 \leq 22$	VERDADERO
$\geq$	Mayor o igual que	$35 \geq 20$	VERDADERO

# 1. Conceptos: Operadores lógicos

Operadores lógicos			
Operador lógico	Jerarquía	Expresión lógica	Significado
NO	(mayor)	No P	NO P No es cierto que P Es FALSO que P
Y	↓	P y Q	$P \wedge Q$ P sin embargo Q
O	(menor)	P o Q	$P \vee Q$ o P o Q o ambas Mínimo P o Q

# 1. Conceptos: Pseudocódigo

Cuando se tiene que resolver un problema de tipo algorítmico, es decir, que tiene una solución determinística, primero se desarrolla el algoritmo, y después, se representa en un pseudocódigo o en un diagrama de flujo.



Un algoritmo determinista que utiliza  $f(n)$  pasos siempre acaba en  $n$  pasos y se obtiene la misma solución. Un algoritmo no determinista que tiene  $f(n)$  niveles no debe devolver el mismo resultado en cada una de sus ejecuciones.



Un **pseudocódigo** es un lenguaje independiente de cualquier lenguaje de programación. Esto es importante porque el solucionador se concentra en la lógica, en las estructuras de datos, en las estructuras de control y no en las reglas de un lenguaje de programación específico.

# 1. Conceptos: Pseudocódigo

Normalmente el **pseudocódigo** no puede ser ejecutado por una computadora, excepto en la actualidad, si se usa una herramienta de simulación como **PSeInt**

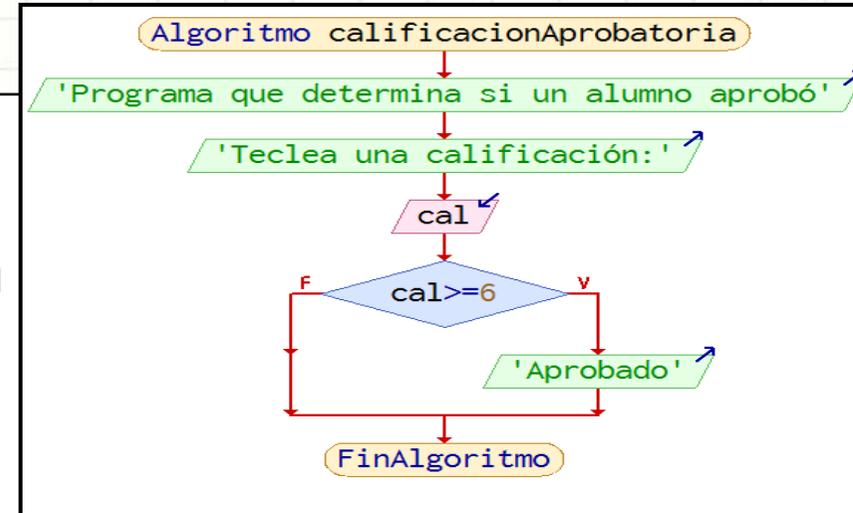
```
Algoritmo CalcularCuadradoCubo
var num, cuadrado, cubo: entero
Inicio
  Escribir "Programa que calcula el
  cuadrado y cubo de un número"
  Escribir "Teclea un numero entero: "
  Leer num
  cuadrado<-num*num
  cubo<-cuadrado*num
  Escribir "El cuadrado es:
  ",cuadrado,"El cubo es: ",cubo
Fin
```



Una vez que domine estas técnicas, podrá codificar fácilmente a un lenguaje de programación como C, Pascal, Java, Python, etc.

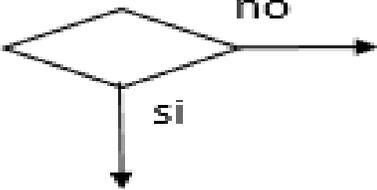
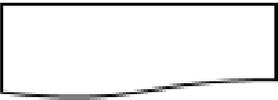
# 1. Conceptos: Diagrama de flujo

Un **diagrama de flujo** al igual que un pseudocódigo son herramientas por computadora que representan algoritmos, es decir, la solución de un problema mediante pasos. Solo que los Diagramas de flujo, son representaciones gráficas.



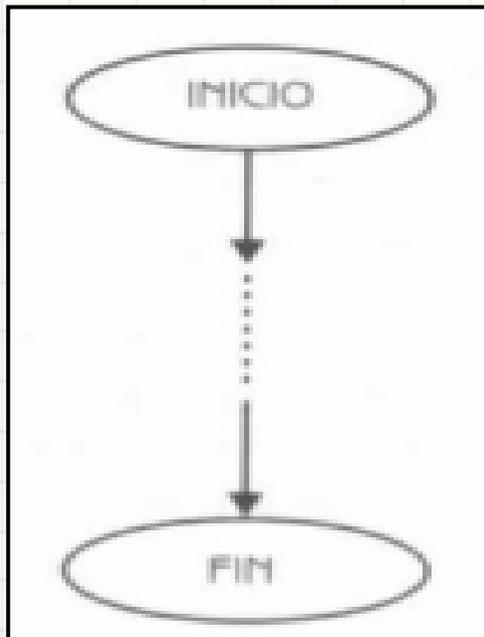
Si un **diagrama de flujo** está completo y correcto, traducirlo a un lenguaje de programación es simple y directo [2].

## Elementos de un diagrama de flujo.

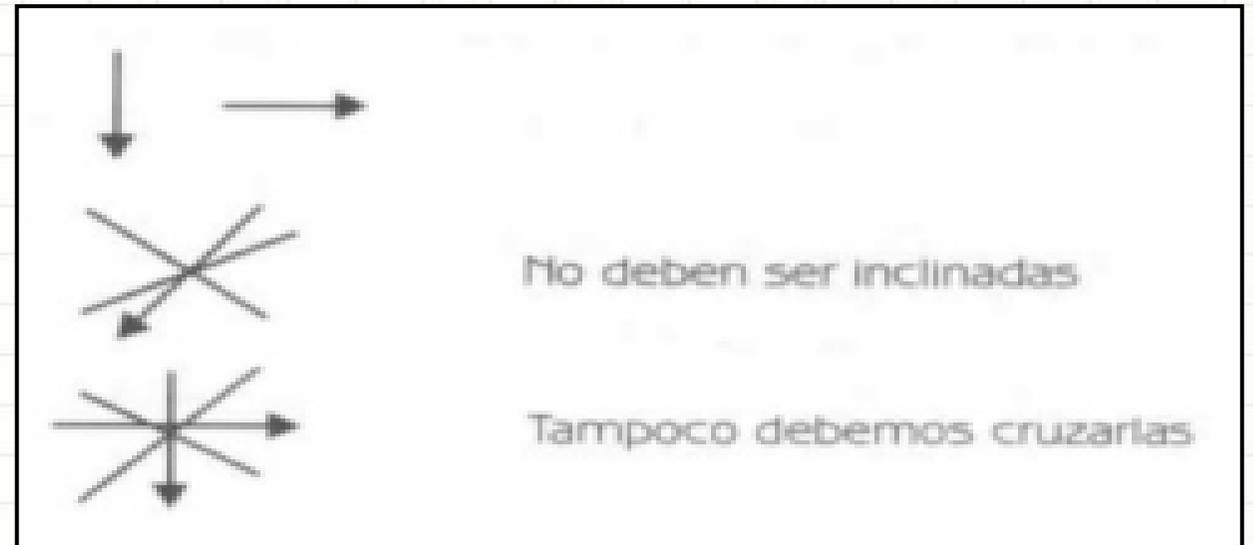
Representación del símbolo	Explicación del símbolo
	Se utiliza para marcar el inicio y el fin del diagrama.
	Se utiliza para introducir datos de entrada. Expresa lectura.
	Representa un proceso. En él se colocan asignaciones, operaciones aritméticas, cambios de valor de celdas en memoria, etc.
	Se utiliza para representar una decisión. En su interior se escribe la condición, dependiendo del resultado de la evaluación sigue el camino verdadero o falso. Este símbolo se utiliza para la estructura de control selectiva o repetitiva mientras o hacer mientras.
	Representa la impresión de un resultado. Es decir, expresa escritura.
	Símbolos utilizados para expresar el flujo del diagrama
	Utilizado para expresar conexión dentro de una misma página.
	Se utiliza para expresar conexión entre páginas diferentes.

# Reglas para la construcción de un diagrama de flujo [2].

1. Todo diagrama de flujo debe tener un inicio y un fin.

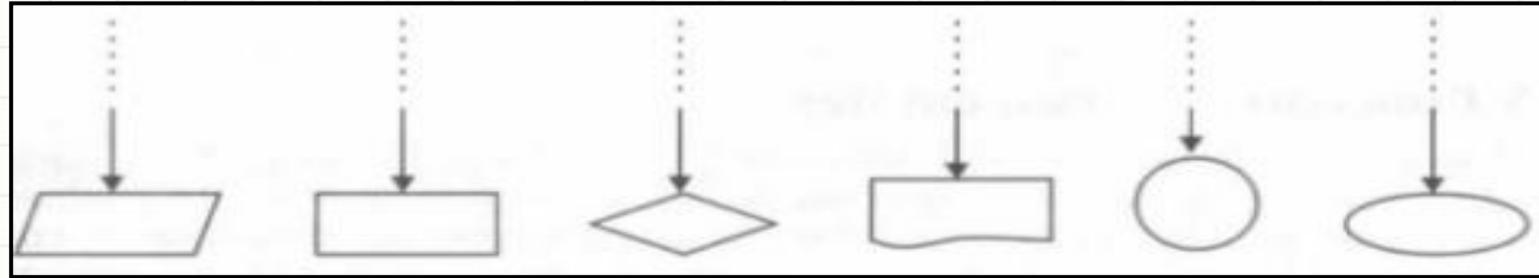


2. Las líneas que indican la dirección del flujo del diagrama deben ser rectas, verticales y horizontales.



## Reglas para la construcción de un diagrama de flujo [2].

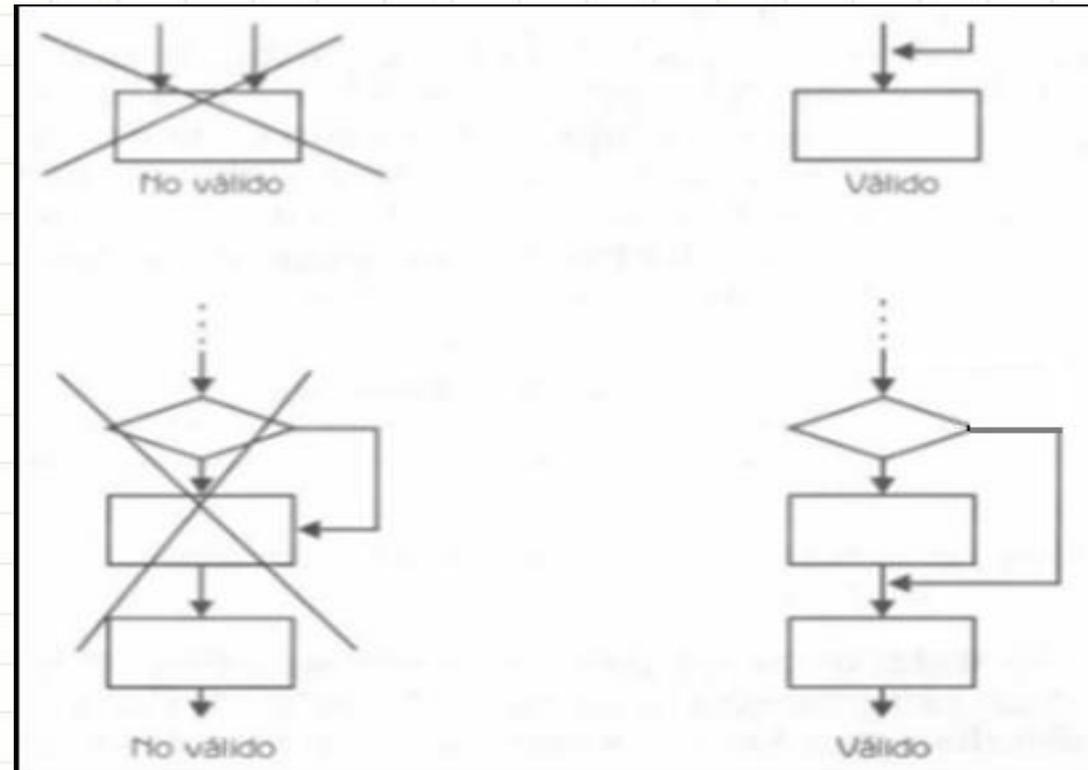
3. Toda línea que indica la dirección del flujo del diagrama debe estar conectada.



4. El diagrama de flujo debe construirse de arriba hacia abajo y de izquierda a derecha
5. La notación utilizada en el diagrama de flujo debe ser independiente del lenguaje de programación.
6. Es conveniente poner comentarios para ayudar a entender lo que hicimos.
7. Se puede utilizar más de una hoja, utilizando los conectores adecuados y numerando las páginas.

## Reglas para la construcción de un diagrama de flujo [2].

8. No pueden llegar más de una línea a un símbolo.



# 1. Conceptos: Bloque de asignación

Un *bloque de asignación* se utiliza para asignar valores o expresiones a una variable.

variable ← expresión o valor

suma ← 0

suma ← suma + 1

La asignación es una operación destructiva. Esto es, si una variable tenía un valor asignado, éste se destruye, conservando el nuevo valor.

Una expresión puede ser aritmética o lógica, o una constante o una variable.



## 2. Estructuras de control

Las *estructuras de control* indican el flujo o camino estándar que debe seguir un proceso de solución de un problema



Se **clasifican** en:  
estructuras de control secuencial,  
selectivas y repetitivas.

## 2. Estructura de control secuencial

La **estructura de control secuencial**, es la más simple de todas. Se utiliza cuando se tiene que ejecutar una instrucción tras otra, incluyendo los bloques de decisión.



Representación en algoritmo.

acción 1

acción 2

...

acción n

Ejemplo1

1. suma  $\leftarrow$  0

2. Escribir "Teclea un numero"

3. Leer num

4. suma  $\leftarrow$  suma + num

**Ejemplo 2.1** A continuación se presenta el primer pseudocódigo. Supongamos que las variables **i**, **acum** y **j** son de tipo **entero**, **rea** y **sum** de tipo **real**, **car** de tipo **caracter** y **band** de tipo **booleano**. Considerar que se tienen que realizar las siguientes asignaciones.

### Algoritmo operacionesAsignación

1.  $i \leftarrow 0$
2.  $i \leftarrow i + 1$
3.  $acum \leftarrow 0$
4.  $j \leftarrow 5^2 / 3$
5.  $car \leftarrow 'a'$
6.  $acum \leftarrow j / i$
7.  $rea \leftarrow acum / 3$
8.  $band \leftarrow (8 > 5) \text{ y } (15 < 2^3)$
9.  $sum \leftarrow acum * 5 / j^2$
10.  $i \leftarrow i * 3$
11.  $rea \leftarrow rea / 5$
12.  $band \leftarrow band \text{ o } (i = j)$
13.  $i \leftarrow rea$
14.  $car \leftarrow j$

FinAlgoritmo

## 2. Estructura de control secuencial

Solo por esta ocasión y en este ejemplo se anota el número de línea u operación de asignación

## 2. Estructura de control secuencial: Prueba de escritorio

### Algoritmo

#### operacionesAsignación

1.  $i \leftarrow 0$
  2.  $i \leftarrow i + 1$
  3.  $acum \leftarrow 0$
  4.  $j \leftarrow 5^2 / 3$
  5.  $car \leftarrow 'a'$
  6.  $acum \leftarrow j / i$
  7.  $rea \leftarrow acum / 3$
  8.  $band \leftarrow (8 > 5) \text{ y } (15 < 2^3)$
  9.  $sum \leftarrow acum * 5 / j^2$
  10.  $i \leftarrow i * 3$
  11.  $rea \leftarrow rea / 5$
  12.  $band \leftarrow band \text{ o } (i = j)$
  13.  $i \leftarrow rea$
  14.  $car \leftarrow j$
- FinAlgoritmo

No. De asignación	i	acum	j	car	rea	band	sum
1	0						
2	1						
3		0					
4			8				
5				'a'			
6		8					
7					2.66		
8						FALSO	
9							0.625
10	3						
11					0.532		
12						FALSO	
13	error						
14				error			

enteros: i,  
acum, j  
caracter: car  
real: rea,  
sum  
boolean:  
band

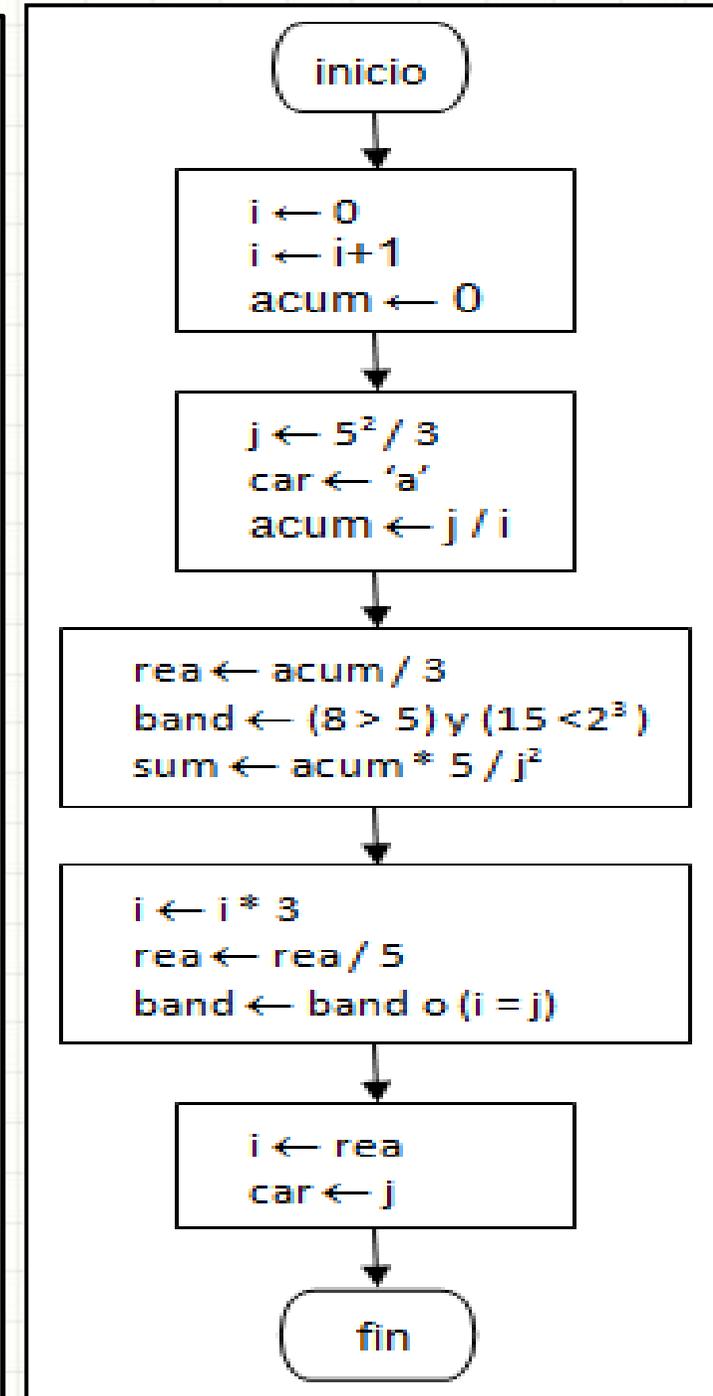
Realiza la prueba de escritorio en tu cuaderno y compara tus resultados.

**Ejemplo 2.1** A continuación se presenta el primer **pseudocódigo** y **su respectivo diagrama de flujo**. Supongamos que las variables  $i$ ,  $acum$  y  $j$  son de tipo **entero**,  $rea$  y  $sum$  de tipo **real**,  $car$  de tipo **caracter** y  $band$  de tipo **booleano**. Considerar que se tienen que realizar las siguientes asignaciones.

Algoritmo operacionesAsignación

1.  $i \leftarrow 0$
2.  $i \leftarrow i + 1$
3.  $acum \leftarrow 0$
4.  $j \leftarrow 5^2 / 3$
5.  $car \leftarrow 'a'$
6.  $acum \leftarrow j / i$
7.  $rea \leftarrow acum / 3$
8.  $band \leftarrow (8 > 5) \text{ y } (15 < 2^3)$
9.  $sum \leftarrow acum * 5 / j^2$
10.  $i \leftarrow i * 3$
11.  $rea \leftarrow rea / 5$
12.  $band \leftarrow band \text{ o } (i = j)$
13.  $i \leftarrow rea$
14.  $car \leftarrow j$

FinAlgoritmo



## 2. Estructura de control secuencial

**Ejemplo 2.2.** Dado por teclado tres caracteres, imprímalos en orden inverso.

Realice el análisis, pseudocódigo, diagrama de flujo y una prueba de escritorio con dos corridas.

## 2. Estructura de control secuencial

**Ejemplo 2.2.** Dado por teclado tres caracteres, imprímalos en orden inverso. Realice el análisis, pseudocódigo, diagrama de flujo y una prueba de escritorio con dos corridas.

**Análisis.**

**Datos de entrada:** ejemplo teclear x, y, z  
variables a,b,c: caracter

**Proceso**

no aplica.

**Salida:**

imprimir z, y, x

Atención con el uso de la sangría o indentación.

A continuación se presenta el segundo **seudocódigo**, donde se identifican algunas **palabras clave**: Algoritmo, Escribir, Leer, Inicio y Fin.

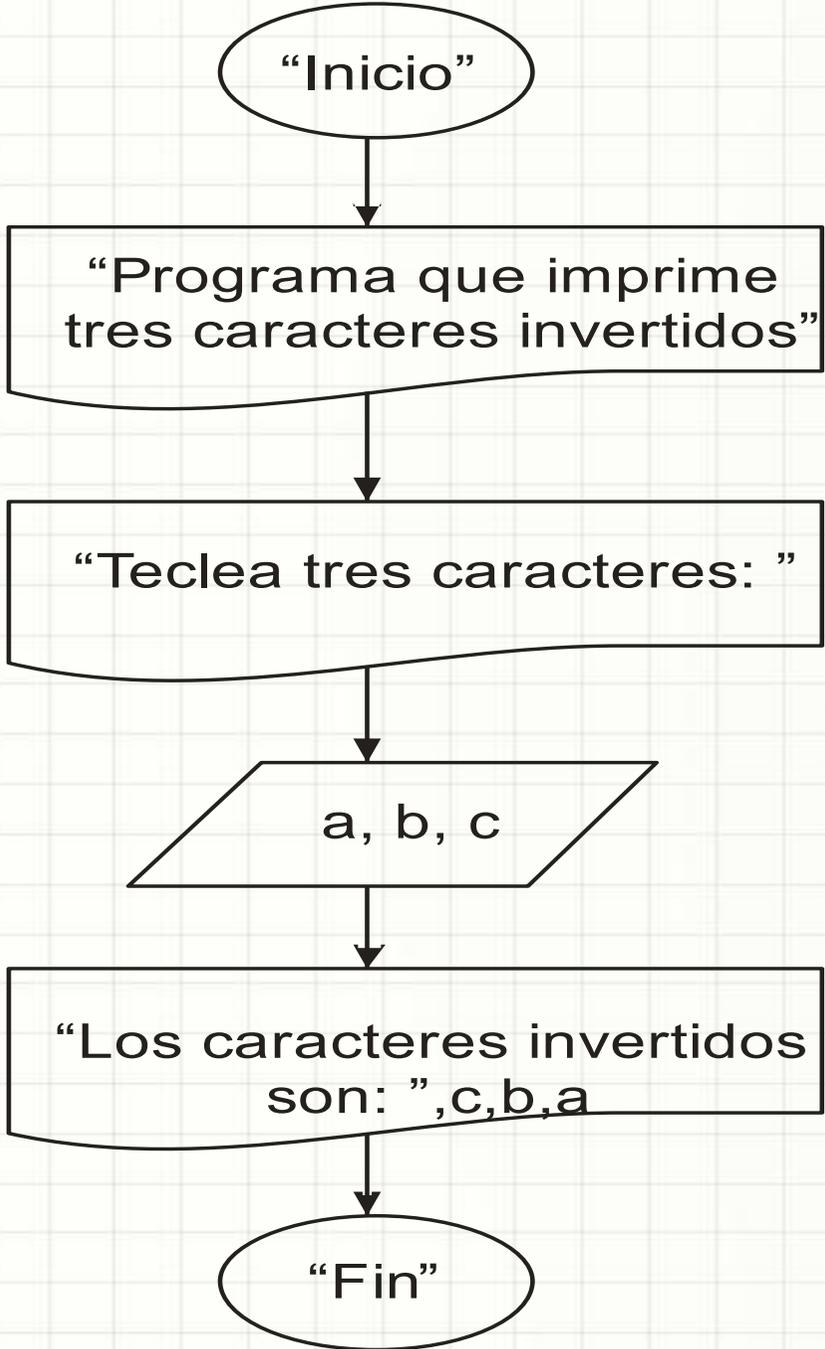
Así mismo, todo pseudocódigo tiene un **nombre**.

```
Algoritmo ImprimirAlReves
var a,b,c: caracter
Inicio
    Escribir "Programa que imprime tres
    caracteres invertidos"
    Escribir "Teclea tres caracteres: "
    Leer a,b,c
    Escribir "Los caracteres invertidos
    son: ",c,b,a
Fin
```

## 2. Estructura de control secuencial

Pseudocódigo y su respectivo Diagrama de flujo

```
Algoritmo ImprimirAlReves
var a,b,c: caracter
Inicio
  Escribir "Programa que imprime tres
  caracteres invertidos"
  Escribir "Teclea tres caracteres: "
  Leer a,b,c
  Escribir "Los caracteres invertidos
  son: ",c,b,a
Fin
```



## 2. Estructura de control secuencial

**Ejemplo 2.3.** Dado por teclado un número entero, calcular e imprimir el cuadrado y el cubo del número.

Realice el análisis, pseudocódigo y una prueba de escritorio con 2 corridas.

**Ejemplo 2.3.** Dado por teclado un número entero, calcular e imprimir el cuadrado y el cubo del número.

Realice el análisis, pseudocódigo y una prueba de escritorio con 2 corridas.

### Análisis.

Datos de entrada: ejemplo se introduce 3.

variables num: **entero**

### Proceso

variables cuadrado, cubo: **entero**

cuadrado  $\leftarrow$  num\*num

cubo  $\leftarrow$  cuadrado\*num

### Salida:

imprimir cuadrado= 9,

cubo = 27

## Pseudocódigo

```
Algoritmo CalcularCuadradoCubo
var num,cuadrado,cubo: entero
Inicio
  Escribir "Programa que calcula el
  cuadrado y cubo de un número"
  Escribir "Teclea un numero entero: "
  Leer num
  cuadrado $\leftarrow$ num*num
  cubo $\leftarrow$ cuadrado*num
  Escribir "El cuadrado es:
  ",cuadrado,"El cubo es: ",cubo
Fin
```

Nota: el carácter  $\leftarrow$  representa la instrucción de asignación( $\leftarrow$ ) vista anteriormente.

## 2. Estructura de control secuencial: Prueba de escritorio

A continuación se presentan dos corridas de pruebas de escritorio del pseudocódigo.

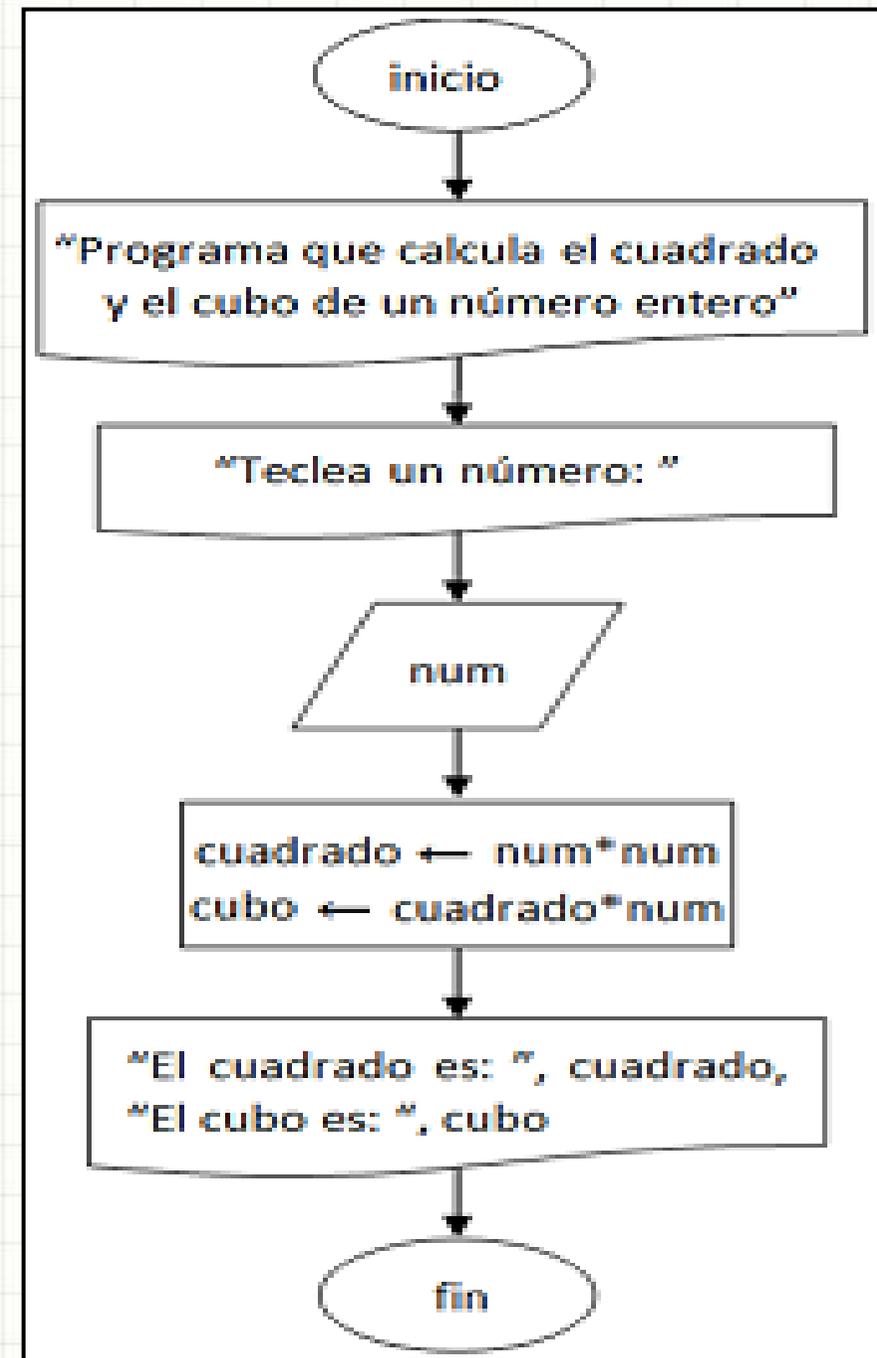
```
Algoritmo CalcularCuadradoCubo
var num, cuadrado, cubo: entero
Inicio
  Escribir "Programa que calcula el
  cuadrado y cubo de un número"
  Escribir "Teclea un numero entero: "
  Leer num
  cuadrado<-num*num
  cubo<-cuadrado*num
  Escribir "El cuadrado es:
  ",cuadrado,"El cubo es: ",cubo
Fin
```

número corrida	num	cuadrado	cubo	salida
1	3	9	27	Programa que calcula el cuadrado y cubo de un número Teclea un número entero: 3 El cuadrado es: 9 El cubo es 27
2	2	4	8	Programa que calcula el cuadrado y cubo de un número Teclea un número entero: 2 El cuadrado es: 4 El cubo es 8

## 2. Estructura de control secuencial

### Pseudocódigo y su respectivo Diagrama de flujo

```
Algoritmo CalcularCuadradoCubo
var num, cuadrado, cubo: entero
Inicio
  Escribir "Programa que calcula el
  cuadrado y cubo de un número"
  Escribir "Teclea un numero entero: "
  Leer num
  cuadrado ← num*num
  cubo ← cuadrado*num
  Escribir "El cuadrado es:
  ", cuadrado, "El cubo es: ", cubo
Fin
```



## 2. Estructura de control secuencial

Para los siguientes ejercicios, desarrollar análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con una corrida.

1. Dado por teclado la matrícula y 4 calificaciones de su primera evaluación, calcule e imprima el promedio y la matrícula.
2. Calcular el área de un trapecio, dado por teclado base mayor, base menor y altura.  $A=(B+b)h/2$ .
3. Dado por teclado el radio y la altura de un cilindro, calcule, e imprima el volumen correspondiente.  $V=(\pi*r^2)h$

## 2. Estructura de control secuencial

4. Dado por teclado una cantidad en pesos y el tipo de cambio de un dólar (ej. 1 dolar equivale a 19 pesos), calcule la conversión e imprima la cantidad correspondiente en dólares.
5. Dado por teclado el peso de un producto en gramos, calcule e imprima el peso en kilogramos.
6. Suponga que le prestan un dinero  $x$  a un porcentaje mensual  $y$ , y que al año solo pagará intereses sin abonar al dinero prestado. Calcule e imprima el total de intereses que tiene que pagar en un año.
7. Le solicitan que vaya a comprar  $x$  kilos de manzana,  $y$  kilos de plátano y  $z$  kilos de mango, al precio por kilo dado por el vendedor. Calcule e imprima el total a pagar.

### 3. Estructura de control selectiva

Esta estructura de control se utiliza en la solución de problemas donde se necesite tomar una decisión.

Esta toma de decisión se basa en la evaluación de una o más condiciones que señalarán como alternativa o consecuencia, la rama a seguir.

En ocasiones la toma de decisiones se realiza en cascada. Es decir, se toma una decisión, se marca la rama correspondiente a seguir, se vuelve a tomar otra decisión y así sucesivamente.



### 3. Estructura de control selectiva

Estas estructuras selectivas se clasifican en :

1. Si Entonces (selectiva simple)
2. Si Entonces / Sino (selectiva doble)
3. Si múltiple (selectiva múltiple)



Cuando a las estructuras selectivas se aplican en cascada, se pueden combinar las estructuras anteriores.

### 3. Estructura de control Selectiva simple (si entonces) [2]

Esta estructura permite que el flujo del programa siga un camino específico **si** se cumple una **condición** o un conjunto de condiciones.

**Si** al evaluar la **condición** (o condiciones) el resultado es verdadero, **entonces** se ejecuta(n) cierta(s) operación(es). Luego se continúa con la secuencia del programa.

Poner especial atención en la **sangría** que se utiliza cuando la condición es verdadera o falsa (ejemplos que siguen).

#### Pseudocódigo

**Si** condición **entonces**

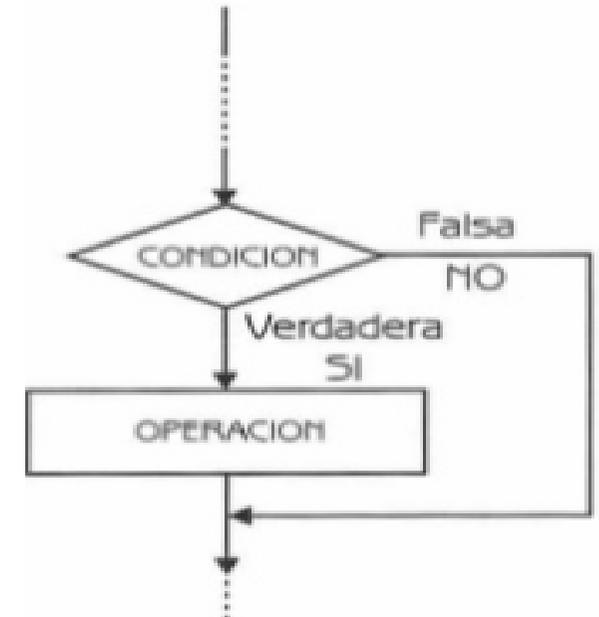
    ejecutar operación 1

    ...

    ejecutar operación n

**FinSi**

#### Diagrama de flujo



### 3. Estructura de control Selectiva simple (si entonces)

**Ejemplo 3.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea mayor o igual que 6.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

**Ejemplo 3.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea mayor o igual que 6.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

### **Análisis del problema**

**Datos entrada:** ejemplo 8.5

Variable cal : real

### **Proceso**

Si  $cal \geq 6$  entonces

    imprimir ' Aprobado '

finSi

### **Salida**

No aplica

**Después, realice otra versión donde la calificación aprobatoria se encuentre entre 6 y 10**

## **3. Selectiva simple (si entonces)**

### **Pseudocódigo**

```
Algoritmo calificacionAprobatoria
var cal: real
Inicio
    Escribir "Programa que
determina si un alumno aprobó"
    Escribir "Teclea una
calificación: "
    Leer cal
    Si  $cal \geq 6$  entonces
        Escribir "Aprobado"
    FinSi
Fin
```

**Repasar operadores relacionales y operadores lógicos [diapositivas 15 y 16]**

### 3. Selectiva simple (si entonces): Prueba de escritorio

```
Algoritmo calificacionAprobatoria
var cal: real
Inicio
  Escribir "Programa que
  determina si un alumno aprobó"
  Escribir "Teclea una
  calificación: "
  Leer cal
  Si cal >= 6 entonces
    Escribir "Aprobado"
  FinSi
Fin
```

Rehacer todos los ejemplos en su cuaderno, incluida la prueba de escritorio para que tenga un mejor entendimiento.

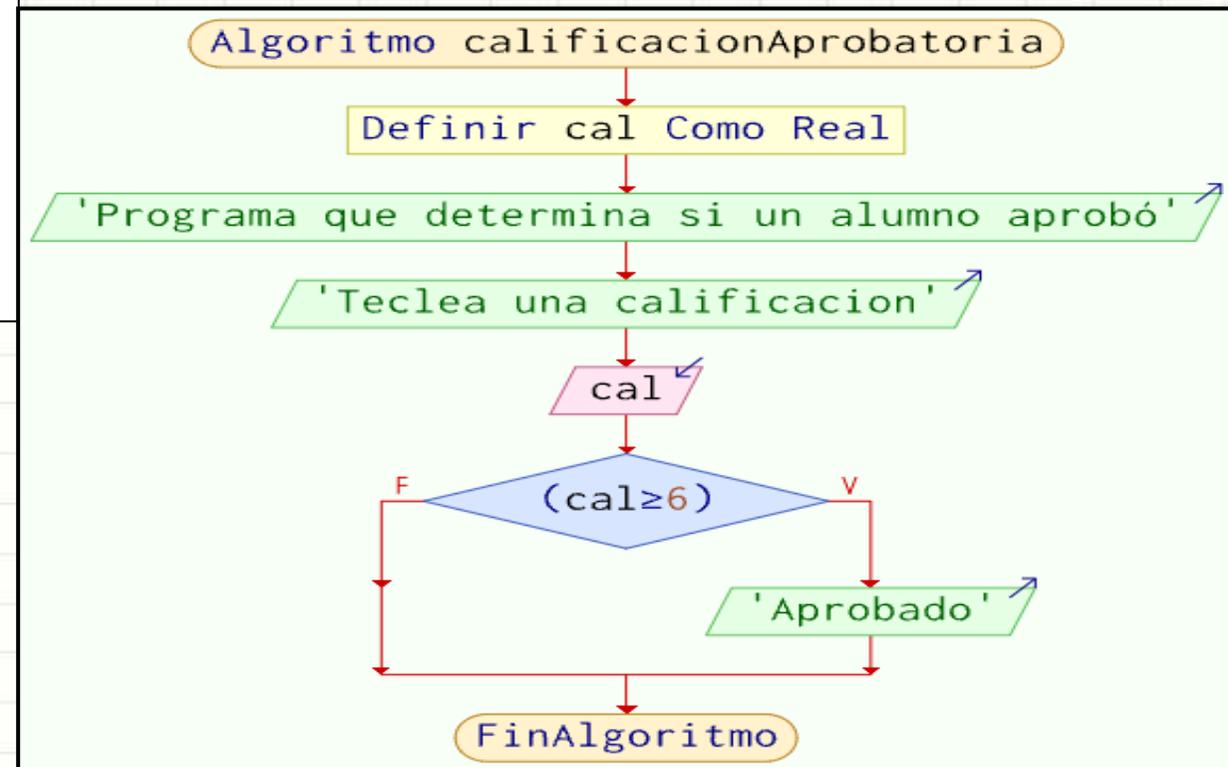
Prueba de escritorio con 2 corridas.

número corrida	cal	cal >= 6	salida
1	7.6	true	Programa que determina ... Teclea un calificación: 7.6 Aprobado
2	5	false	Programa que determina ... Teclea un calificación: 5

### 3. Selectiva simple (si entonces)

```
Algoritmo calificacionAprobatoria
var cal: real
Inicio
  Escribir "Programa que
  determina si un alumno aprobó"
  Escribir "Teclea una
  calificación: "
  Leer cal
  Si cal >= 6 entonces
    Escribir "Aprobado"
  FinSi
Fin
```

### Pseudocódigo y su respectivo Diagrama de flujo



### 3. Selectiva simple (si entonces)

**Ejemplo 3.2.** Considere como entrada el sueldo de un trabajador, calcule un aumento del 15% si su sueldo es menor a \$1000, e imprima en este caso el nuevo sueldo.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

### 3. Selectiva simple (si entonces)

**Ejemplo 3.2.** Considere como entrada el sueldo de un trabajador, calcule un aumento del 15% si su sueldo es menor a \$1000, e imprima en este caso el nuevo sueldo.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

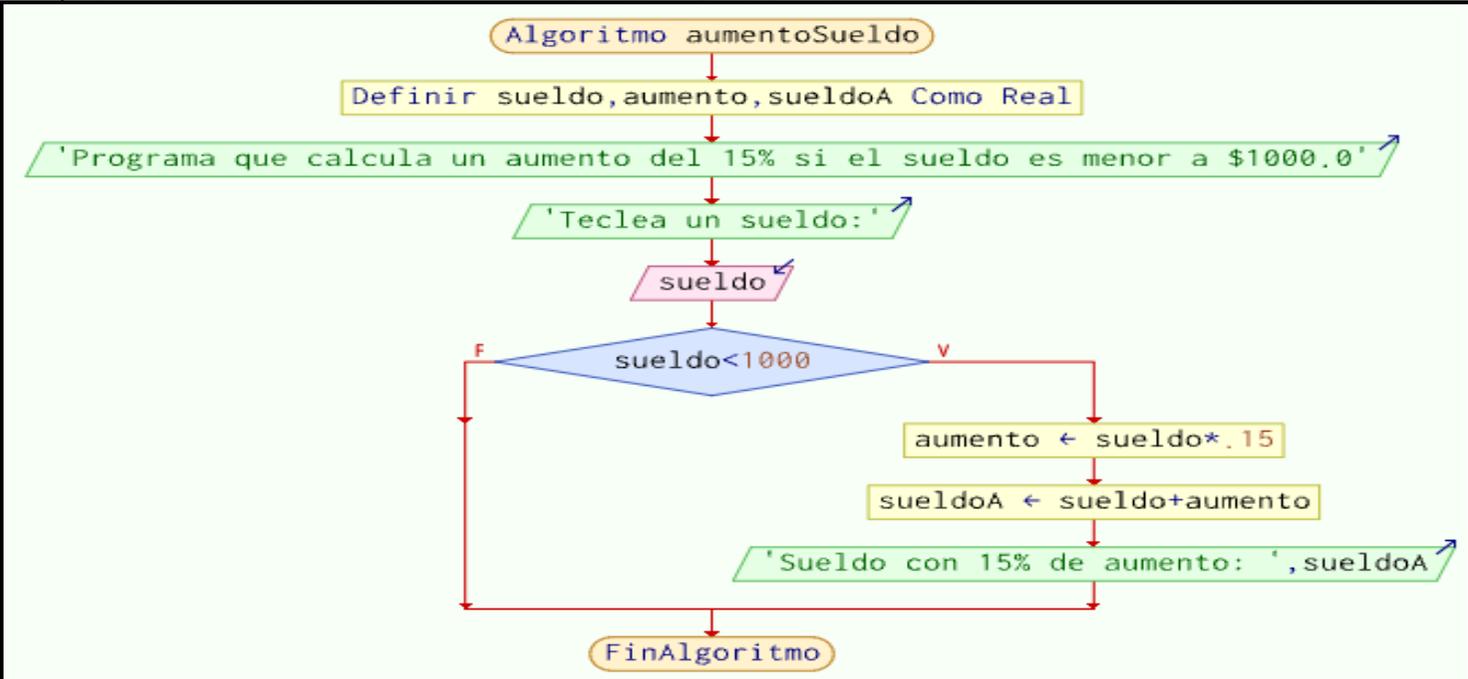
#### Pseudocódigo

```
Algoritmo aumentoSueldo
var sueldo,aumento,sueldoA: real
Inicio
    Escribir "Programa que calcula un
    aumento del 15% si el sueldo es
    menor a $1000.0"
    Escribir "Teclea el sueldo: "
    Leer sueldo
    Si sueldo<1000 entonces
        aumento<-sueldo*.15
        sueldoA<-sueldo+aumento
        Escribir "Sueldo con aumento:
        ", sueldoA
    FinSi
Fin
```

### 3. Selectiva simple (si entonces)

```
Algoritmo aumentoSueldo
var sueldo, aumento, sueldoA: real
Inicio
  Escribir "Programa que calcula un
  aumento del 15% si el sueldo es
  menor a $1000.0"
  Escribir "Teclea el sueldo: "
  Leer sueldo
  Si sueldo<1000 entonces
    aumento<-sueldo*.15
    sueldoA<-sueldo+aumento
    Escribir "Sueldo con aumento:
    ", sueldoA
  FinSi
Fin
```

**Pseudocódigo y su respectivo Diagrama de flujo**



## 4. Selectiva doble (si entonces / sino) [2]

Esta estructura de control permite que el flujo del algoritmo **se bifurque por dos ramas diferentes** en el punto de la toma de decisión.

Si al evaluar la condición, resulta verdadera, entonces se sigue un camino específico y se ejecutan ciertas operaciones. En caso que la condición sea falsa, se sigue otro camino y se ejecutan otras operaciones.

### Pseudocódigo

**Si** condición **entonces**

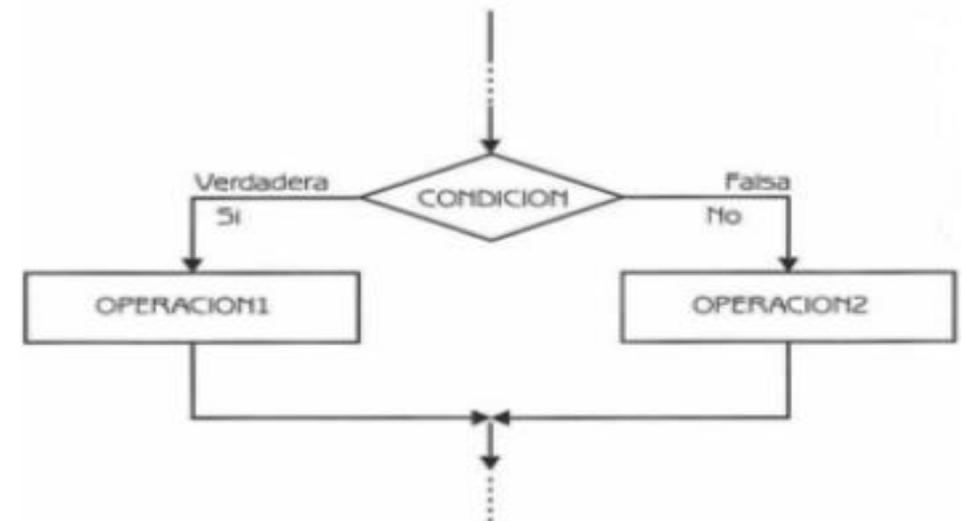
ejecutar operación 1  
ejecutar operación n

**Sino**

ejecutar operación sino 1  
ejecutar operación sino m

**FinSi**

### Diagrama de flujo



## 4. Selectiva doble (si entonces / sino)

**Ejemplo 4.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación se encuentre en el rango de 6 a 10. En otro caso, imprima "reprobado".

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con 2 corridas.*

## 4. Selectiva doble (si entonces / sino)

**Ejemplo 4.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea de 6 a 10, en otro caso, imprima "reprobado".

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con 2 corridas.*

### Análisis del problema

**Datos entrada:** ejemplo 9.2

Variable cal : real

### Proceso

**Si** cal $\geq$ 6 y cal $\leq$ 10 **entonces**

Imprimir "Aprobado"

**Sino**

Imprimir 'Reprobado'

**finSi**

### Salida

No aplica

Uso de  
**sangría**

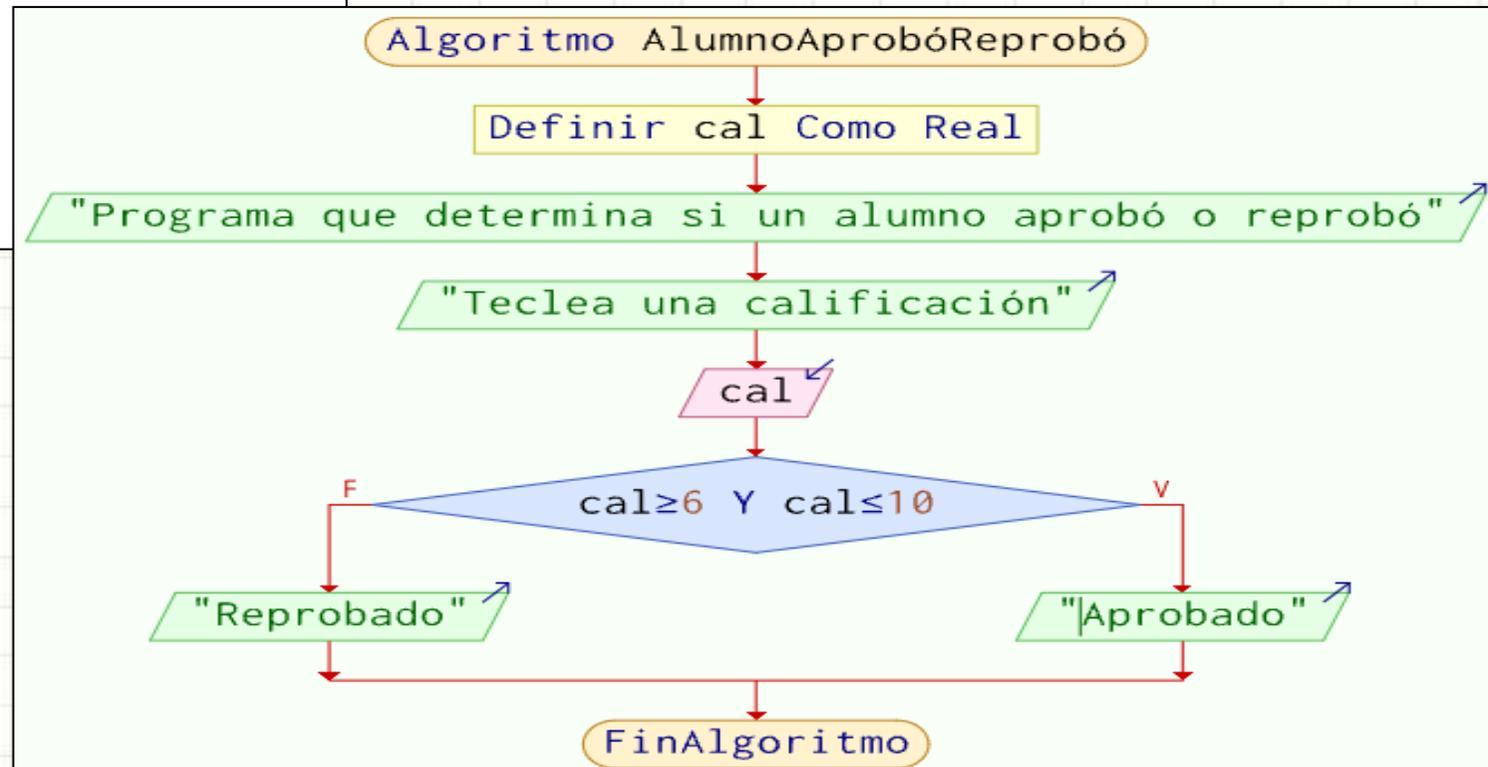
### Pseudocódigo

```
Algoritmo alumnoAprobóOReprobó
var cal: real
Inicio
    Escribir "Programa que determina si
    un alumno aprobó o reprobó"
    Escribir "Teclea una calificación: "
    Leer cal
    Si cal $\geq$ 6 y cal  $\leq$ 10 entonces
        Escribir "Aprobado"
    Sino
        Escribir "Reprobado"
    FinSi
Fin
```

## 4. Selectiva doble (si entonces / sino)

```
Algoritmo alumnoAprobóOReprobó
var cal: real
Inicio
  Escribir "Programa que determina si
  un alumno aprobó o reprobó"
  Escribir "Teclea una calificación: "
  Leer cal
  Si cal >= 6 y cal <= 10 entonces
    Escribir "Aprobado"
  Sino
    Escribir "Reprobado"
  FinSi
Fin
```

**Pseudocódigo y su respectivo  
Diagrama de flujo**



## 4. Selectiva doble (si entonces / sino)

**Ejemplo 4.2.** Dado un sueldo de un trabajador por teclado, calcule un aumento del 15% si su sueldo es menor a \$1000 y 12% en caso contrario. Imprima el nuevo sueldo del trabajador.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

## 4. Selectiva doble (si entonces / sino)

**Ejemplo 4.2.** Dado un sueldo de un trabajador por teclado, calcule un aumento del 15% si su sueldo es menor a \$1000 y 12% en caso contrario. Imprima el nuevo sueldo del trabajador.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

### Pseudocódigo

```
Algoritmo aumentoSueldo
var sueldo, aumento, sueldoA: real
Inicio
    Escribir 'Programa que calcula un
    aumento del 15% si el sueldo es menor
    a $1000.0 y 12% en caso contrario'
    Escribir 'Teclea un sueldo: '
    Leer sueldo
    Si sueldo<1000 entonces
        aumento<-sueldo*.15
    Sino
        aumento<-sueldo*.12
    FinSi
    sueldoA<-sueldo+aumento
    Escribir 'Sueldo con aumento: ',
    sueldoA
Fin
```

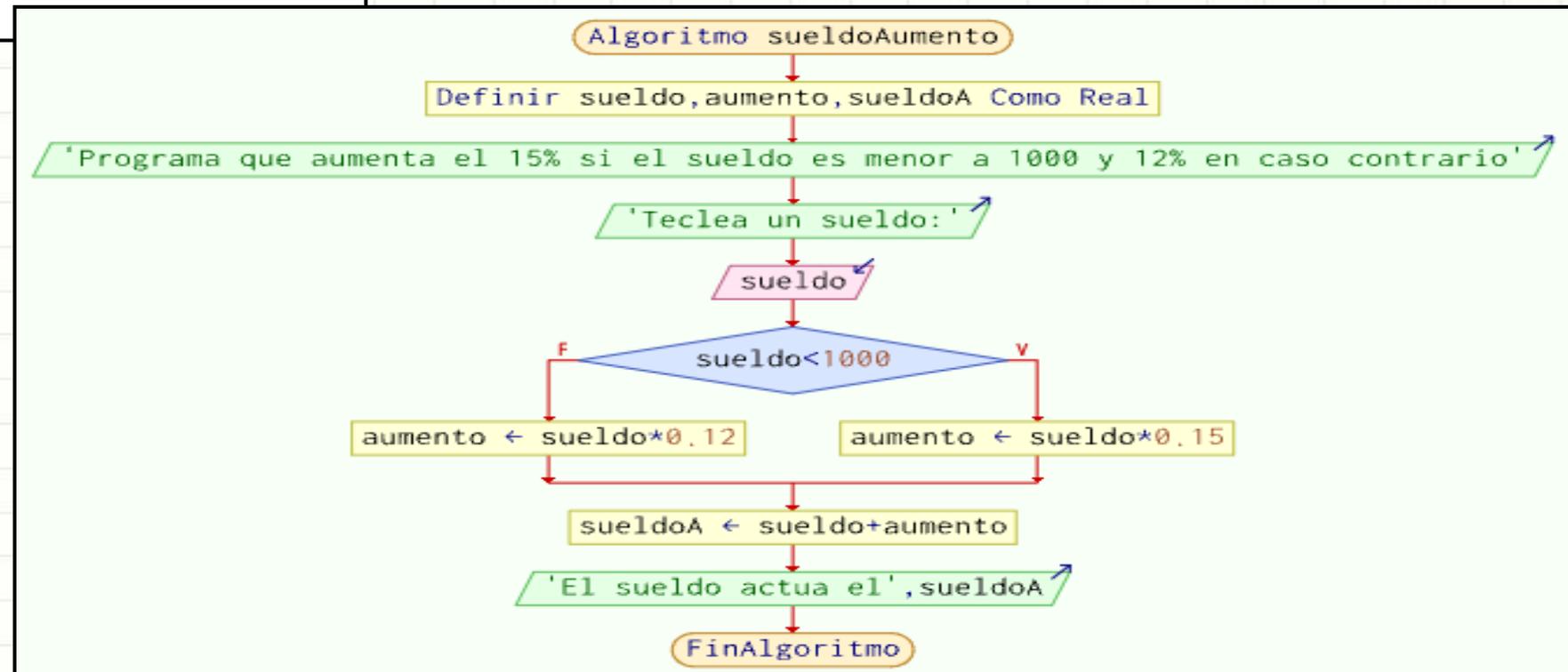
```

Algoritmo aumentoSueldo
var sueldo, aumento, sueldoA: real
Inicio
  Escribir 'Programa que calcula un
  aumento del 15% si el sueldo es menor
  a $1000.0 y 12% en caso contrario'
  Escribir 'Teclea un sueldo: '
  Leer sueldo
  Si sueldo<1000 entonces
    aumento<-sueldo*.15
  Sino
    aumento<-sueldo*.12
  FinSi
  sueldoA<-sueldo+aumento
  Escribir 'Sueldo con aumento: ',
  sueldoA
Fin

```

## 4. Selectiva doble (si entonces / sino)

### Pseudocódigo y su respectivo Diagrama de flujo



## 4. Selectiva doble (si entonces / sino)

**Ejemplo 4.3.** Dado tres números por teclado, encontrar e imprimir el mayor de ellos.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

## 4. Selectiva doble (si entonces / sino)

### Pseudocódigo

**Ejemplo 4.3.** Dado tres números por teclado, encontrar e imprimir el mayor de ellos.

*Realice el análisis, pseudocódigo, diagrama de flujo, y prueba de escritorio con 2 corridas.*

```
Algoritmo mayorTresNumeros
var num1, num2, num3, nMayor: enteros
Inicio
.....
Escribir 'Programa que encuentra el
mayor de tres números'
Escribir 'Teclea tres números'
Leer num1, num2, num3
Si num1>num2 entonces
.....
    nMayor <- num1
Sino
.....
    nMayor <- num2
FinSi
Si num3 > nMayor entonces
.....
    nMayor <- num3
FinSi
Escribir 'El mayor es: ', nMayor
Fin
```

## 5. Selectiva múltiple (según sea) [2]

Permite que el flujo del algoritmo se **bifurque por varias ramas** en el punto de la toma de decisión, esto en función del valor que tome el **selector** (variable entera o carácter). Así, si el **selector** toma: el valor 1, se ejecutará la acción 1 ó acciones en ese camino, si toma el valor 2, se ejecutará la acción 2 ó acciones en ese camino, si toma el valor N se realizará la acción N ó acciones en ese camino. Después se continuará con el flujo normal del algoritmo, realizándose la acción N+1

...

**SegunSea** selector

**caso 1:** ejecutar acción 1

**caso 2:** ejecutar acción 2

...

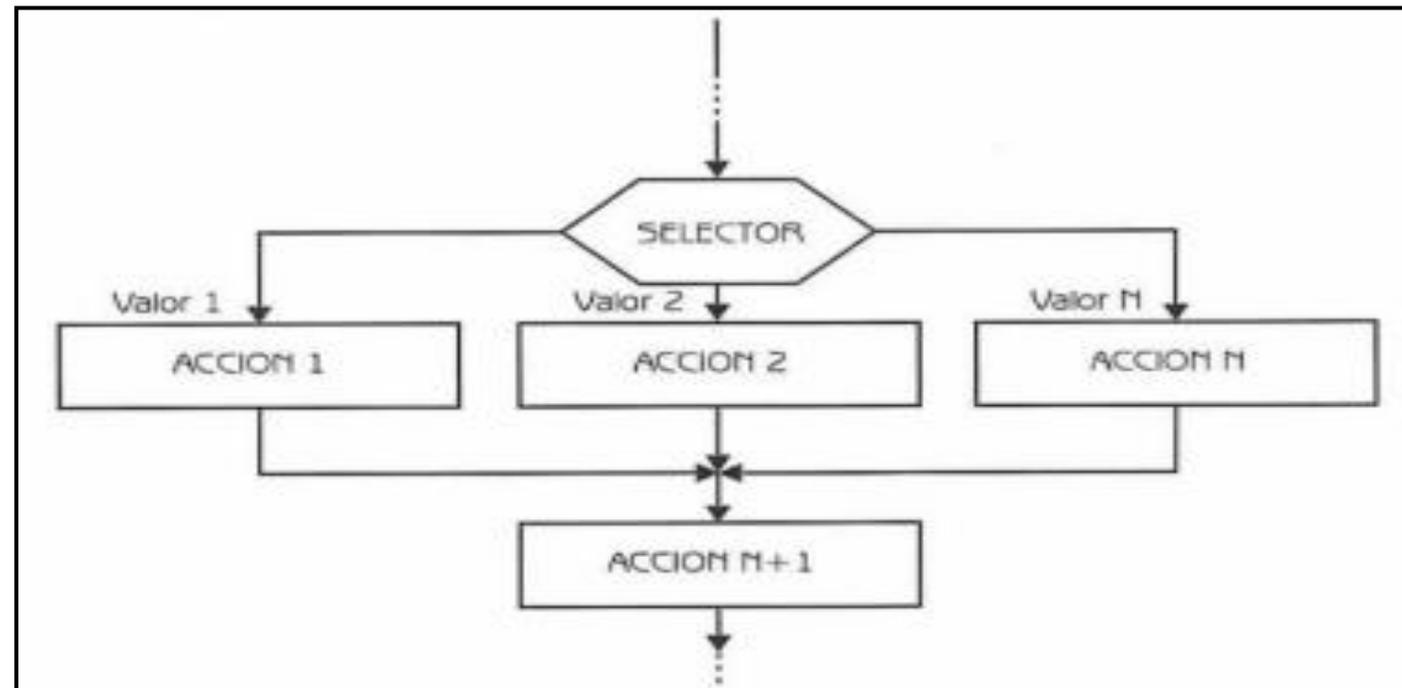
**caso N:** ejecutar acción N

**SinOpcion:**

ejecutar acción sinOpción 1

**FinSegunSea**

Hacer acción N+1



## 5. Selectiva múltiple (según sea)

**Ejemplo 5.1.** Dado por teclado una categoría y el sueldo de un trabajador, calcule el aumento correspondiente de acuerdo a la siguiente tabla. Finalmente, imprima el porcentaje de aumento y el nuevo sueldo.

Categoría	Aumento
1	15%
2	10%
3	8%
4	7%

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con al menos 2 corridas.*

## 5. Selectiva múltiple (SegúnSea)

### Pseudocódigo versión 1

**Ejemplo 5.1.** Dado por teclado una categoría y el sueldo de un trabajador, calcule el aumento correspondiente de acuerdo a la siguiente tabla. Finalmente, imprima el porcentaje de aumento y el nuevo sueldo.

Categoría	Aumento
1	15%
2	10%
3	8%
4	7%

```
Algoritmo aumentoSueldoCategoría
var categ: entero
    sueldo, nsueldo: real
Inicio
    Escribir "Programa que dada la categoría y
sueldo, calcula el aumento correspondiente"
    Escribir "Teclea la categoría[1-4]: "
    Leer categ
    Escribir "Teclea un sueldo: "
    Leer sueldo
    nsueldo<-sueldo;
    Escribir "Le corresponde aumento de "
    SegúnSea categ
        Caso 1: Escribir "15%"
                nsueldo <- sueldo*1.15
        Caso 2: Escribir "10%"
                nsueldo <- sueldo*1.10
        Caso 3: Escribir "8%"
                nsueldo <- sueldo*1.08
        Caso 4: Escribir "7%"
                nsueldo <- sueldo*1.07
    SinOpcion:
        Escribir "Categoría no válida"
    FinSegúnSea
    Escribir "Su nuevo sueldo es de: ", nsueldo
Fin
```

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con al menos 2 corridas.*

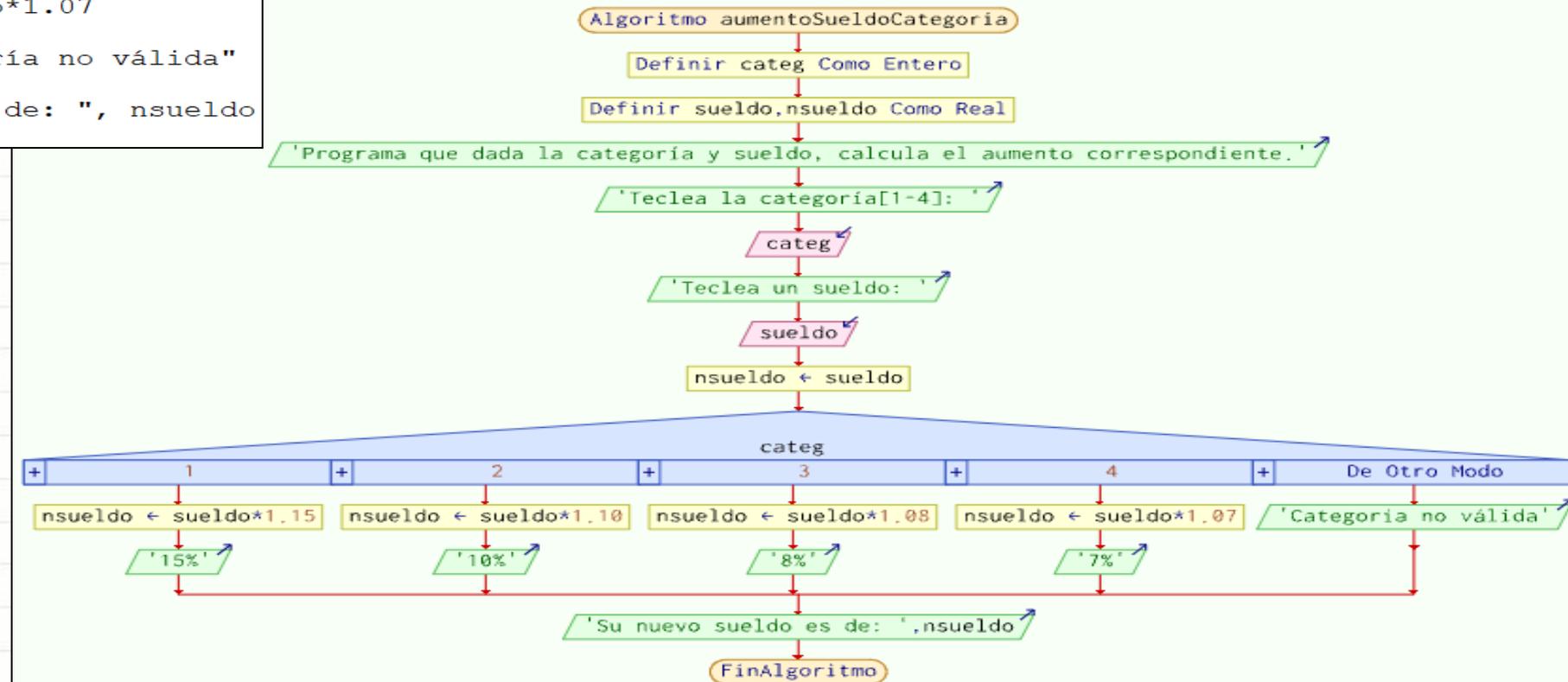
```

Algoritmo aumentoSueldoCategoria
var categ: entero
    sueldo, nsueldo: real
Inicio
Escribir "Programa que dada la categoría y
sueldo, calcula el aumento correspondiente"
Escribir "Teclea la categoría[1-4]: "
Leer categ
Escribir "Teclea un sueldo: "
Leer sueldo
nsueldo<-sueldo;
Escribir "Le corresponde aumento de "
SegunSea categ
    Caso 1: Escribir "15%"
        nsueldo <- sueldo*1.15
    Caso 2: Escribir "10%"
        nsueldo <- sueldo*1.10
    Caso 3: Escribir "8%"
        nsueldo <- sueldo*1.08
    Caso 4: Escribir "7%"
        nsueldo <- sueldo*1.07
SinOpcion:
    Escribir "Categoría no válida"
FinSegunSea
Escribir "Su nuevo sueldo es de: ", nsueldo
Fin

```

## 5. Selectiva múltiple (SegúnSea)

### Pseudocódigo y su respectivo Diagrama de flujo



## 5. Selectiva múltiple (SegúnSea)

**Ejemplo 5.1.** Dado por teclado una categoría y el sueldo de un trabajador, calcule el aumento correspondiente, de acuerdo a la siguiente tabla. Finalmente, imprima tanto por ciento y su nuevo sueldo.

Categoría	Aumento
1	15%
2	10%
3	8%
4	7%

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con al menos 2 corridas.*

### Pseudocódigo versión 2

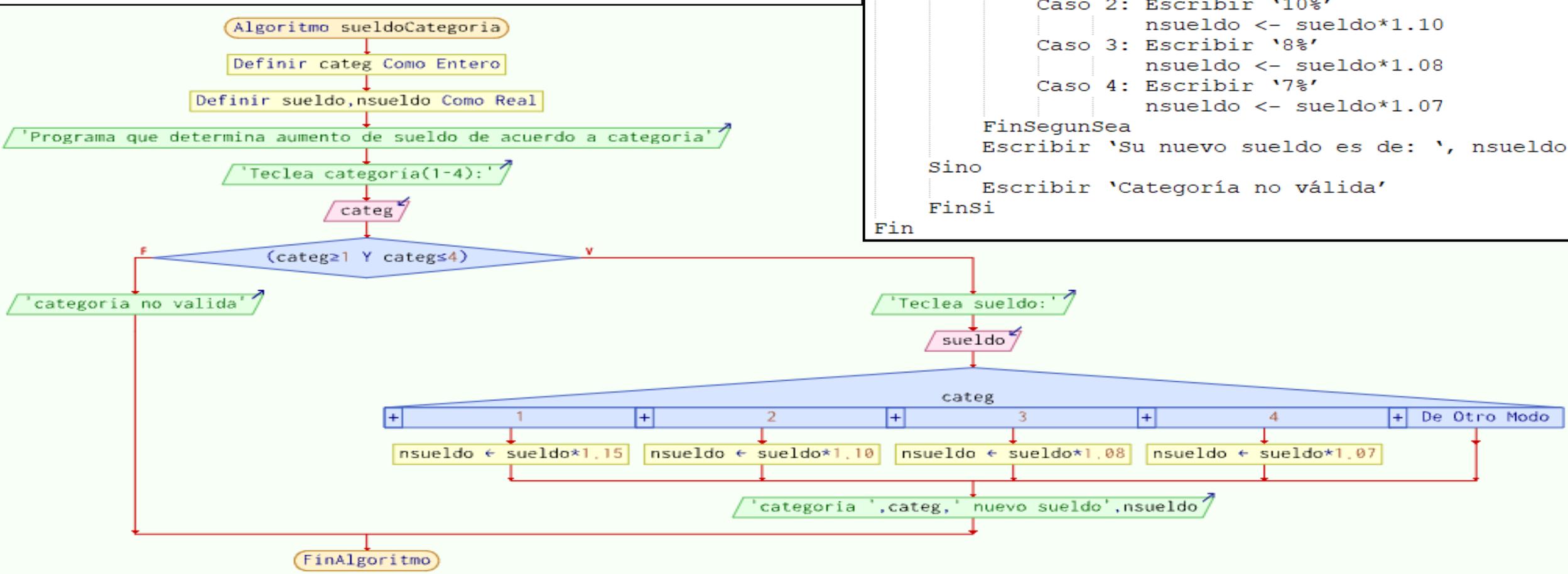
```
Algoritmo aumentoSueldoCategoría
var categ: entero
    sueldo, nsueldo: real
Inicio
    Escribir 'Programa que dada la categoría y
    sueldo, calcula el aumento correspondiente'
    Escribir 'Teclea la categoría: '
    Leer categ
    Si categ >=1 y categ <=4 entonces
        Escribir 'Teclea un sueldo: '
        Leer sueldo
        Escribir 'Le corresponde aumento de '
        SegúnSea categ
            Caso 1: Escribir '15%'
                    nsueldo <- sueldo*1.15
            Caso 2: Escribir '10%'
                    nsueldo <- sueldo*1.10
            Caso 3: Escribir '8%'
                    nsueldo <- sueldo*1.08
            Caso 4: Escribir '7%'
                    nsueldo <- sueldo*1.07
        FinSegúnSea
        Escribir 'Su nuevo sueldo es de: ', nsueldo
    Sino
        Escribir 'Categoría no válida'
    FinSi
Fin
```

# 5. Selectiva múltiple (SegúnSea)

## Pseudocódigo versión 2 y su respectivo Diagrama de flujo

```

Algoritmo aumentoSueldoCategoria
var categ: entero
    sueldo, nsueldo: real
Inicio
    Escribir 'Programa que dada la categoría y
    sueldo, calcula el aumento correspondiente'
    Escribir 'Teclea la categoría: '
    Leer categ
    Si categ >=1 y categ <=4 entonces
        Escribir 'Teclea un sueldo: '
        Leer sueldo
        Escribir 'Le corresponde aumento de '
        SegúnSea categ
            Caso 1: Escribir '15%'
                nsueldo <- sueldo*1.15
            Caso 2: Escribir '10%'
                nsueldo <- sueldo*1.10
            Caso 3: Escribir '8%'
                nsueldo <- sueldo*1.08
            Caso 4: Escribir '7%'
                nsueldo <- sueldo*1.07
        FinSegúnSea
        Escribir 'Su nuevo sueldo es de: ', nsueldo
    Sino
        Escribir 'Categoría no válida'
    FinSi
Fin
    
```

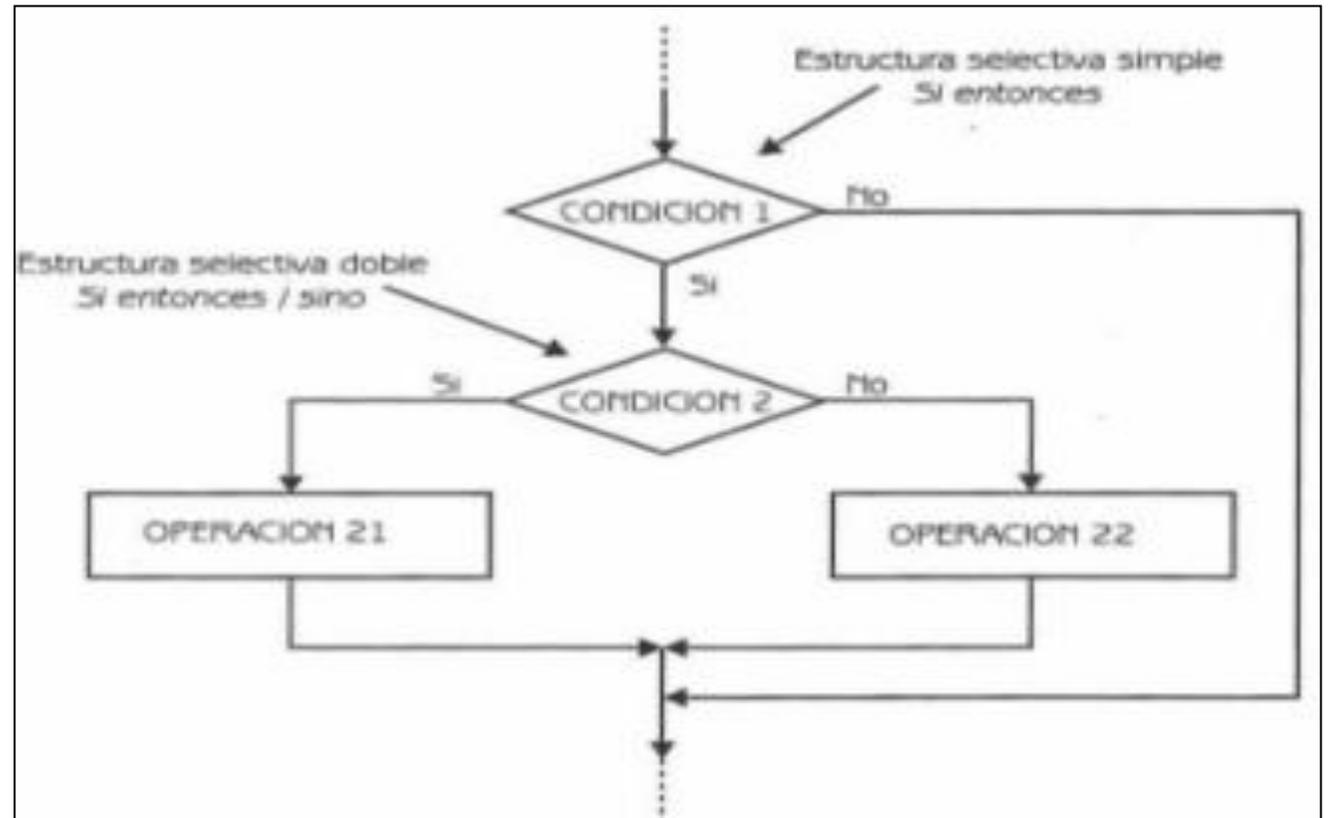


## 5. Selectivas en cascada (anidadas) [2]

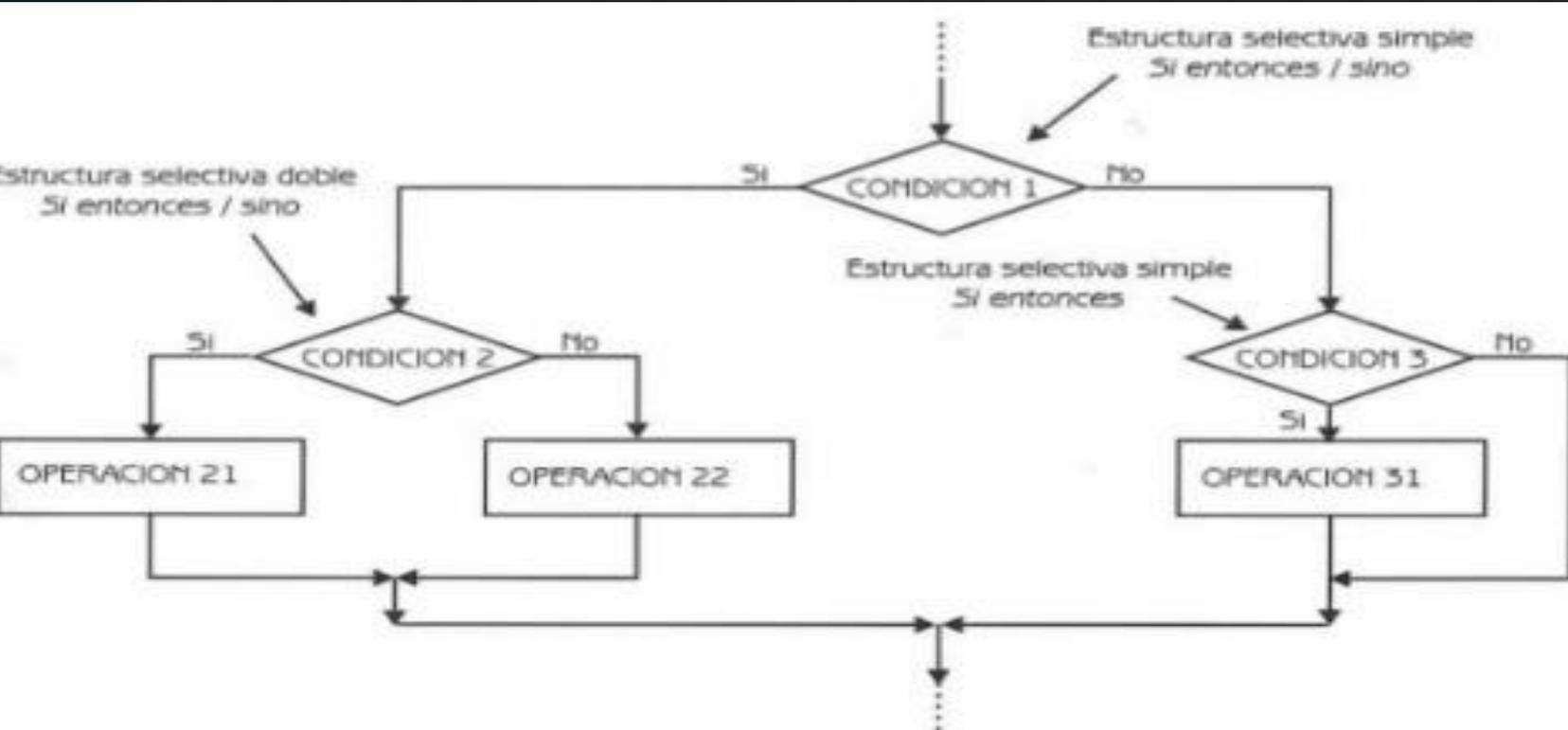
En muchos casos en la solución de problemas, después de tomar una decisión y marcar el camino a seguir, es necesario tomar otra decisión y así sucesivamente.

Selectiva doble

```
...  
Si condición1 entonces  
    Si condición2 entonces  
        Acción 21  
    Sino  
        Acción 22  
FinSi  
Sino  
    Acción 12  
FinSi  
...
```



## 5. Selectivas en cascada (anidadas) [2]



```
...  
Si condición1 entonces  
    Si condición2 entonces  
        Acción 21  
    Sino  
        Acción 22  
    FinSi  
Sino  
    Si condición3 entonces  
        Acción 31  
    FinSi  
FinSi  
...
```

## 5. Selectivas en cascada (anidadas) [2]

```
...
Si condición1 entonces
  SegunSea selector
    Caso 1: Acción 21
    Caso 2:
      Si condición3 entonces
        Acción 31
      FinSi
    SinOpcion:
      Si condición4 entonces
        Acción 41
      Sino
        Accion42
      FinSi
    FinSegunSea
  Sino
    Si condición5 entonces
      Acción 51
    FinSi
  FinSi
...
```

Selectiva doble

Selectiva múltiple

Selectiva simple

Selectiva doble

Selectiva simple



## 5. Selectivas en cascada (anidadas)

**Ejemplo 5.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea de 6 a 10, en otro caso si la calificación es de 0 y menor que 6, imprima "reprobado".

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con 2 corridas.*

## 5. Selectivas en cascada (anidadas)

**Ejemplo 5.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea de 6 a 10, en otro caso si la calificación es de 0 y menor que 6, imprima "reprobado".

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con 2 corridas.*

Recuerde aplicar **sangría** a todos sus programas correctamente

### Pseudocódigo

```
Algoritmo calificacionAprobatoria
var cal: real
Inicio
    Escribir 'Programa que determina si
un alumno aprobó'
    Escribir 'Teclea una calificación: '
    Leer cal
    Si cal >= 6 y cal <= 10 entonces
        Escribir 'Aprobado'
    Sino
        Si cal >= 0 y cal < 6 entonces
            Escribir 'Reprobado'
        FinSi
    FinSi
Fin
```

## 5. Selectivas en cascada (anidadas)

### Pseudocódigo

```
Algoritmo calificacionAprobatoria
var cal: real
Inicio
    Escribir 'Programa que determina si
un alumno aprobó'
    Escribir 'Teclea una calificación: '
    Leer cal
    Si cal >= 6 y cal <= 10 entonces
        Escribir 'Aprobado'
    Sino
        Si cal >= 0 y cal < 6 entonces
            Escribir 'Reprobado'
        FinSi
    FinSi
Fin
```

Realizar su diagrama de flujo y posteriormente su prueba de escritorio

## 5. Selectivas en cascada (anidadas)

**Ejemplo 5.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea de 6 a 10, en otro caso si la calificación es de 0 y menor que 6, imprima "reprobado", **en caso contrario, "Calificación no válida"**.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con 2 corridas.*

## 5. Selectivas en cascada (anidadas)

**Ejemplo 5.1.** Dada una calificación por teclado, escriba "aprobado" en caso que la calificación sea de 6 a 10, en otro caso si la calificación es de 0 y menor que 6, imprima "reprobado", **en caso contrario, "Calificación no válida"**.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con 2 corridas.*

### Pseudocódigo

```
Algoritmo alumnoAprobóReprobó
var cal: real
Inicio
    Escribir 'Programa que determina si un
    alumno aprobó o reprobó'
    Escribir 'Teclea una calificación: '
    Leer cal
    Si cal >= 6 y cal <= 10 entonces
        Escribir 'Aprobado'
    Sino
        Si cal >= 0 y cal < 6 entonces
            Escribir 'Reprobado'
        Sino
            Escribir 'Calificación no válida'
        FinSi
    FinSi
Fin
```

# 5. Selectivas en cascada (anidadas)

## Pseudocódigo

```
Algoritmo alumnoAprobóReprobó
var cal: real
Inicio
  Escribir 'Programa que determina si un
  alumno aprobó o reprobó'
  Escribir 'Teclea una calificación: '
  Leer cal
  Si cal >= 6 y cal <= 10 entonces
    Escribir 'Aprobado'
  Sino
    Si cal >= 0 y cal < 6 entonces
      Escribir 'Reprobado'
    Sino
      Escribir 'Calificación no válida'
    FinSi
  FinSi
Fin
```

Realizar su diagrama de flujo y posteriormente su prueba de escritorio

## 5. Selectivas en cascada (anidadas)

**Ejemplo 5.2** Dado por teclado tres números enteros diferentes, imprimir los números en orden descendente.

*Para el siguiente Pseudocódigo, realice su diagrama de flujo. Después realice las siguientes corridas de pruebas de escritorio: 8,1,7; 2,5,9; 2,8,5.*

*Por último escriba un pseudocódigo con otro camino de solución.*

## 5. Selectivas en cascada (anidadas).

**Ejemplo 5.2** Dado por teclado tres números enteros diferentes, imprimir los números en orden descendente.

*Para el siguiente Pseudocódigo, realice su diagrama de flujo.*

*Después realice las siguientes corridas de pruebas de escritorio: 8,1,7; 2,5,9; 2,8,5.*

*Por último escriba un pseudocódigo con otro camino de solución.*

### Pseudocódigo

```
Algoritmo numerosOrdenDescendente
var num1,num2,num3: enteros
Inicio
  Escribir 'Programa que dado tres
  números enteros diferentes los imprime
  en orden descendente'
  Escribir 'Teclea tres números: '
  Leer num1, num2, num3
  Si num1>num2 entonces
    Si num2>num3 entonces
      Escribir num1, num2, num3
    Sino
      Si num1>num3 entonces
        Escribir num1, num3, num2
      Sino
        Escribir num3, num1, num2
      FinSi
    FinSi
  Sino
    Si num1>num3 entonces
      Escribir num2, num1, num3
    Sino
      Si num3>num2 entonces
        Escribir num3, num2, num1
      Sino
        Escribir num2, num3, num1
      FinSi
    FinSi
  FinSi
Fin
```

## Ejercicios de estructuras de control selectivas

Para cada uno de los siguientes ejercicios realice su pseudocódigo, diagrama de flujo, y prueba de escritorio con dos corridas.

1. Dado por teclado como datos los valores enteros P y Q, determine si los mismos satisfacen la siguiente expresión:

$$P^3 + Q^4 - 2 * P^2 < 680$$

2. Dado como entrada un número entero, determine e imprima si el número es positivo o negativo.

## Ejercicios de estructuras de control selectivas

3. Dado por teclado un número determine e imprima si es par o impar, para esto apóyese de la operación módulo que obtiene el residuo de la operación.
4. Dado como entrada la matrícula y 5 calificaciones de un alumno, calcule e imprima el promedio y la palabra "aprobado" si el alumno tiene un promedio mayor o igual que 6, y la palabra "no aprobado" en caso contrario.
5. Dado por teclado tres números enteros, determine si están en orden decreciente.

## Ejercicios de estructuras de control selectivas

6. Dado como entrada el valor de los coeficientes  $a, b, c$  de la expresión que resuelve una ecuación de segundo grado  $ax^2 + bx + c = 0$ , encuentre el valor de las raíces  $x_1, x_2$ , mediante el cálculo del discriminante  $D=b^2-4ac$ , si el discriminante es  $\geq 0$  imprima el resultado de las raíces, en otro caso, imprimir "las raíces son imaginarias".

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

## Ejercicios de estructuras de control selectivas

7. En la compra de artículos, dependiendo del monto le aplican un descuento:

- Si el monto es menor que \$500, no hay descuento.
- Si el monto se encuentra entre \$500 y menor que \$1000, el 5%.
- Si el monto se encuentra entre \$1000 y menor que \$7000, el 11%.
- Si el monto se encuentra entre \$7000 y menor que \$15000, el 18%.
- Si el monto es mayor o igual a \$15000, el 25%.

## Ejercicios de estructuras de control selectivas

8. Realizar las operaciones básicas con 2 números leídos por teclado. Considere que la división por cero es una indeterminación.
9. Dado como entrada un número del 1 al 7, imprima el correspondiente día de la semana (Ej.: 1. Lunes; 2. Martes; etc). Si ingresa un número fuera del rango, deberá imprimir el mensaje "Error... valor fuera de rango".
10. Dado por teclado dos números reales y el símbolo de una operación básica (+, -, \*, /), imprima el resultado correspondiente. Si el símbolo no es correcto, imprima el mensaje "Error en símbolo".

# Referencias

1. Zapata Ospina, Carlos A. "Fundamentos de programación, Guía de autoenseñanza". Alfaomega Ra-Ma. 2006
2. Cairó Osvaldo. "Metodología de la programación". Alfaomega, 3ª edición. 2005.
3. Joyanes Aguilar, Luis. "Fundamentos de programación: Algoritmos, estructura de datos y objetos". McGraw-Hill, 4ª edición. 2008