



# **INSTITUTO TECNOLÓGICO BELTRÁN**

Centro de Tecnología e Innovación



Algoritmos de programación – Lic. Yaps David P.

## ANÁLISIS DE UN PROBLEMA

Identifique las conclusiones (o suposiciones) iniciales para la resolución del problema si cada letra equivale a un valor numérico distinto de un solo dígito. Luego, determine el valor que recibe cada una de ella.

$$\begin{array}{r} \phantom{+} \text{D} \text{O} \text{S} \\ + \text{D} \text{O} \text{S} \\ \hline \text{T} \text{R} \text{E} \text{S} \end{array}$$

## ANÁLISIS DE UN PROBLEMA

$$\begin{array}{r} + \text{ D O S} \\ \text{D O S} \\ \hline \text{T R E S} \end{array}$$

D	O	S	T	R	E

El valor de S debe ser un número capaz de que la suma de si mismo, del el mismo valor

El valor de O debe ser distinto de D ya que arrojan distintos resultados

El valor de D debe ser mayor o igual a 5, ya que el resultado agrega un dígito adicional

## ANÁLISIS DE UN PROBLEMA

$$\begin{array}{r} \text{+ D O S} \\ \text{D O S} \\ \hline \text{T R E S} \end{array}$$

D	O	S	T	R	E
7	3	0	1	4	6

$$\begin{array}{r} \text{+ 7 3 0} \\ \text{7 3 0} \\ \hline \text{1 4 6 0} \end{array}$$

## VARIABLES

Son contenedores de información, donde cada una de ellas tiene asociado un nombre único que lo identifica y puede almacenar solo un determinado tipo de dato y, para algunos de ellos, es necesario determinar una longitud (siempre que lo sepamos con anterioridad).

Los tipos de datos básicos que utilizaremos son numéricos, textos, booleanos y tipos especiales. Asimismo:

Siguiendo el ejemplo analizado anteriormente, para las variables D, O, S, T, R, E; las definiremos como el tipo numérico y longitud de 1.

## PSEUDOCÓDIGO

El pseudocódigo es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

Utiliza las convenciones estructurales de un lenguaje de programación real, pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación.

Se utiliza pseudocódigo pues este es más fácil de entender para las personas que el código del lenguaje de programación convencional, ya que es una descripción eficiente y con un entorno independiente de los principios fundamentales de un algoritmo. Se utiliza en la planificación del desarrollo de programas informáticos, para esbozar la estructura del programa antes de realizar la efectiva codificación.

Un programador que tiene que aplicar un algoritmo específico, generalmente comienza con una descripción en pseudocódigo, y luego traduce esa descripción en el lenguaje de programación en el que se encuentra desarrollando y lo modifica para que interactúe correctamente con el resto del programa.

Los programadores también pueden iniciar un proyecto describiendo la forma del código en pseudocódigo en el papel antes de escribirlo en su lenguaje de programación.

## OPERADOR DE ASIGNACIÓN

Es aquel que nos permite realizar la carga (asignación) de un determinado valor a una variable. Se asigna lo especificado en el lado derecho de la sentencia.

Símbolo	Descripción	Ejemplo
<- =	Asignación simple	A <- 1 A = 1

En una asignación se pueden utilizar el contenido de otra variable e, inclusive, la misma:

Descripción	Ejemplo
Asignación del tipo contador	variable <- variable + 1 variable = variable + 1
Asignación del tipo acumulador	variable1 <- variable1 + variable2 variable1 = variable1 + variable2

## OPERADORES ARITMETICOS

Tal como era de esperarse los operadores aritméticos comprenden las cuatro operaciones básicas: suma, resta, multiplicación y división; con un agregado: el operador módulo.

Símbolo	Descripción	Ejemplo
+	Suma	$A + B$
-	Resta	$A - B$
*	Multiplicación	$A * B$
/	División	$A / B$
%	Módulo o resto de una división	$A \% B$
-	Signo	$-A$

El operador módulo ( % ) se utiliza para calcular el resto del cociente entre dos enteros.



## OPERADORES ARITMETICOS

Para evitar errores en los cálculos se pueden usar paréntesis, sin limitación de anidamiento, los que fuerzan a realizar primero las operaciones incluidas en ellos. Los paréntesis no disminuyen la velocidad a la que se ejecuta el programa sino que tan sólo obligan al compilador a realizar las operaciones en un orden dado, por lo que es una buena costumbre utilizarlos ampliamente.

Los paréntesis tienen un orden de precedencia 0, es decir que antes que nada se evalúa lo que ellos encierran.

El último de los operadores aritméticos es el de *signo*. No debe confundírsele con el de resta, ya que este es un operador unitario que opera sobre una única variable cambiando el signo de su contenido numérico. Obviamente no existe el operador + unitario, ya que su operación sería DEJAR el signo de la variable, lo que se consigue simplemente por omisión del signo.

## OPERADORES RELACIONALES

Todas las operaciones relacionales dan sólo dos posibles resultados: verdadero o falso.

Símbolo	Descripción	Ejemplo
<	Menor que	(A < B)
>	Mayor que	(A > B)
<=	Menor o igual que	(A <= B)
>=	Mayor o igual que	(A >= B)
==	Igual que	(A == B)
<>	distinto	(A <> B)

Uno de los errores más comunes es confundir el operador relacional igual que (==) con el de asignación igual a (=). La expresión a=b copia el valor de b en a, mientras que a == b retorna un cero, si a es distinto de b o un número distinto de cero si son iguales.

## MÉTODO O ESQUEMA DE RESOLUCIÓN

Será una etapa más a cumplir antes de empezar a trabajar en su programación.

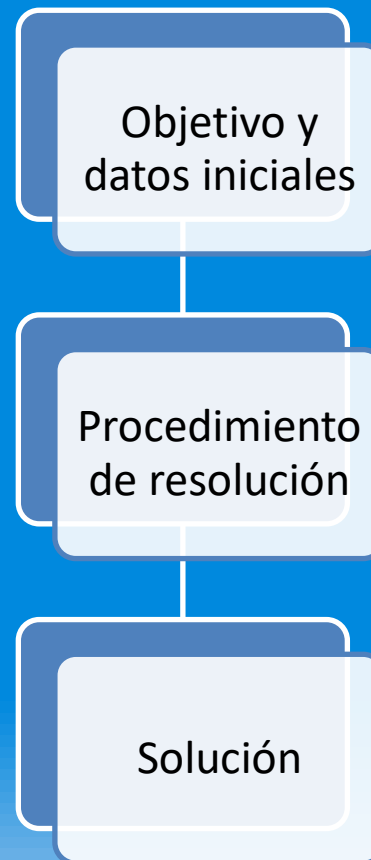
Los métodos aplicables son muy diversos, pero haremos una clasificación práctica de distintas formas o tipos de resolución de problemas a los que nos enfrentaremos como programadores.

Un programa extenso puede ser una mezcla de distintos tipos de problema.



## PROBLEMA CON RESOLUCIÓN DIRECTA

Se trataría de todo tipo de problemas que solucionamos mentalmente, de forma sencilla, en uno o varios pasos.



## PROBLEMA CON RESOLUCIÓN DIRECTA

Se trataría de todo tipo de problemas que solucionamos mentalmente, de forma sencilla, en uno o varios pasos.

Calcular la superficie de un cuadrado, dada la longitud de uno de sus lados

$$\text{Superficie} = \text{Lado} * \text{Lado}$$

Mostrar  
Superficie

**Mostrar** 'Ingrese longitud de lado'

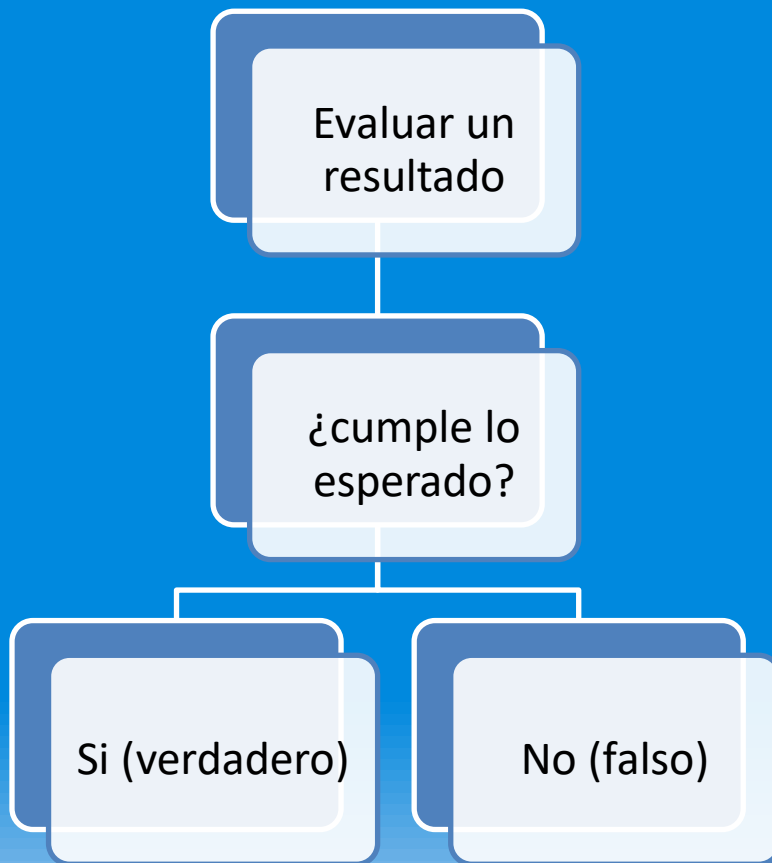
**Leer** Lado

Superficie <- Lado \* Lado

**Mostrar** Superficie

## PROBLEMA CON BIFURCACIONES

Llega un momento en que necesitamos hacer una bifurcación condicional: si se cumple una condición hacer una cosa, si no se cumple hacer otra. Para ello colocaremos las llamadas bifurcaciones.



**Si** Condición **Entonces**  
acciones\_por\_verdadero

**Sino**  
acciones\_por\_falso

**Fin Si**

## TABLAS DE VERDAD

Son un elemento de la lógica proposicional para determinar el valor de verdad (es decir, si es verdadero o falso) de una proposición. Los valores de verdad posibles son dos: verdadero y falso, que también pueden expresarse como 1 y 0. Estos valores se llaman booleanos por el álgebra de Boole, que tiene la particularidad de operar con datos binarios.

Normalmente se representa al valor Verdadero con la letra V y al valor Falso con la F.

Una proposición es una afirmación capaz de tener un valor de verdad. Es decir, una oración de la cual se puede decir que es verdadera o que es falsa.

«EL DÍA ESTÁ SOLEADO»

(será verdadero o será falso, según si el día está o no está soleado).

## PROPOSICIÓN

Una proposición puede ser: atómica si no puede subdividirse, o molecular si está compuesta por dos o más proposiciones, unidas por un operador lógico.

Proposición atómica:  
«EL DÍA ESTÁ SOLEADO»

Proposición molecular:  
«EL DÍA ESTÁ SOLEADO Y CALUROSO»

Es molecular porque puede subdividirse en dos proposiciones: «el día está soleado» y «el día está caluroso».



# OPERACIONES LÓGICAS

Una operación lógica se compone de operandos (proposiciones) y operadores.

Mediante una operación lógica se unen proposiciones para obtener una nueva proposición compuesta.

Los operadores lógicos usados para unir proposiciones son:

AND (Y - Conjunción)

OR (O - Disyunción)

NOT (NO - Negación)

Mediante ellos se forman proposiciones moleculares.

## TABLA DE VERDAD DE CADA OPERADOR LÓGICO

### CONJUNCIÓN

P	Q	<i>Y</i>
V	V	V
V	F	F
F	V	F
F	F	F

### DISYUNCIÓN

P	Q	<i>O</i>
V	V	V
V	F	V
F	V	V
F	F	F

### NEGACIÓN

P	<i>NO</i>
V	F
F	V

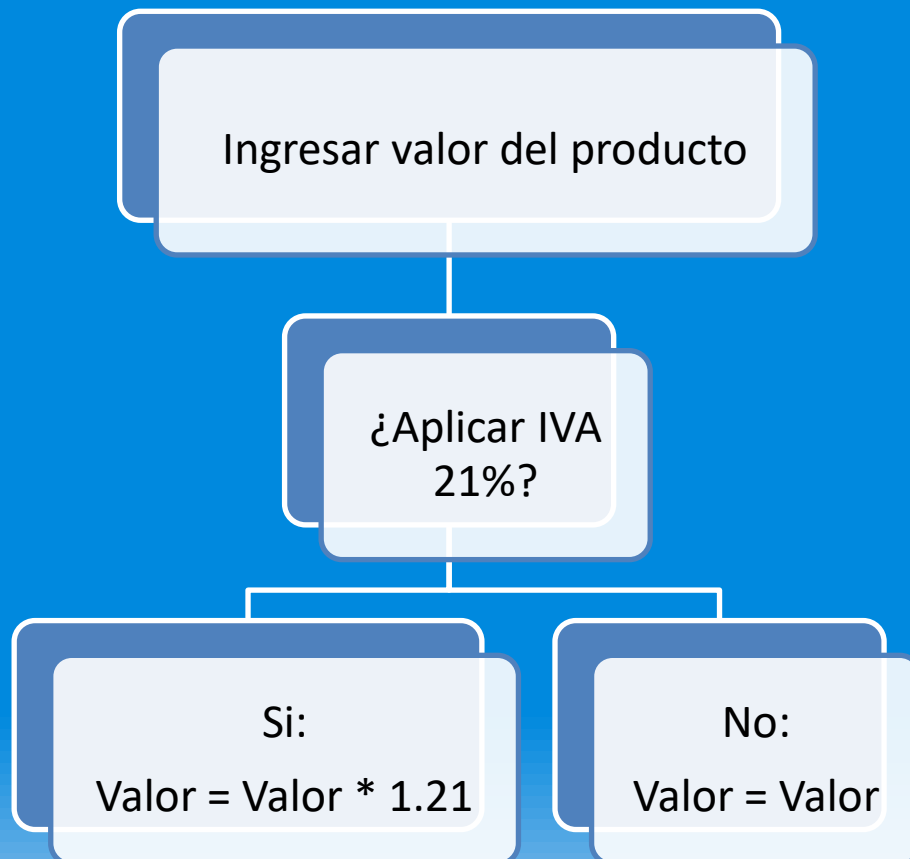
## OPERADORES LÓGICOS

Hay tres operadores que realizan las conectividades lógicas Y (AND) , O (OR) y NEGACION (NOT). Los resultados de la operaciones lógicas siempre adoptan los valores cierto o falso. La evaluación de las operaciones lógicas se realiza de izquierda a derecha y se interrumpe cuando se ha asegurado el resultado. El operador negación invierte el sentido lógico de las operaciones

Símbolo	Descripción	Ejemplo
Y	Y (AND)	(A>B) && (C<D)
O	O (OR)	(A>B)    (C<D)
NO	NEGACIÓN (NOT)	!(A>B)

## PROBLEMA CON BIFURCACIONES

Llega un momento en que necesitamos hacer una bifurcación condicional: si se cumple una condición hacer una cosa, si no se cumple hacer otra. Para ello colocaremos las llamadas bifurcaciones.



**Mostrar** 'Valor del producto'

**Leer** Valor

**Mostrar** '¿Aplicar IVA 21%?'

**Leer** AplicarIVA

**Si** 'S' = AplicarIVA **Entonces**  
Valor <- Valor \* 1.21

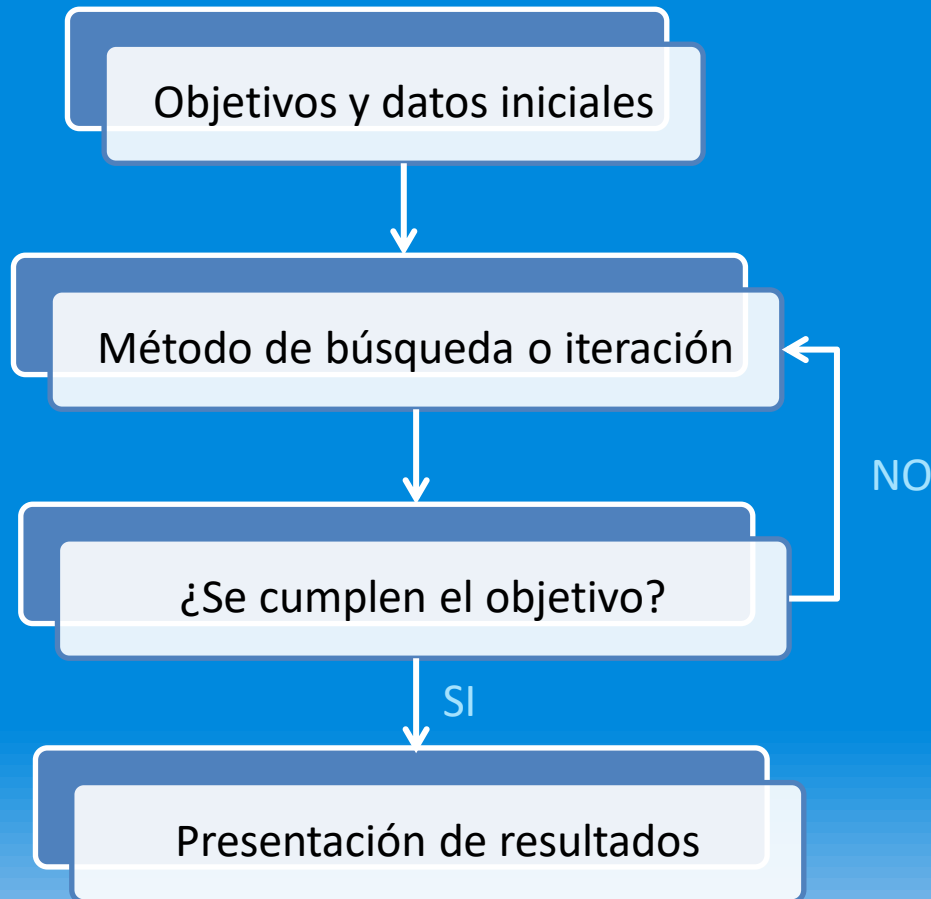
**Sino**  
Valor <- Valor

**Fin Si**

**Mostrar** Valor

## PROBLEMA CON ITERACIONES

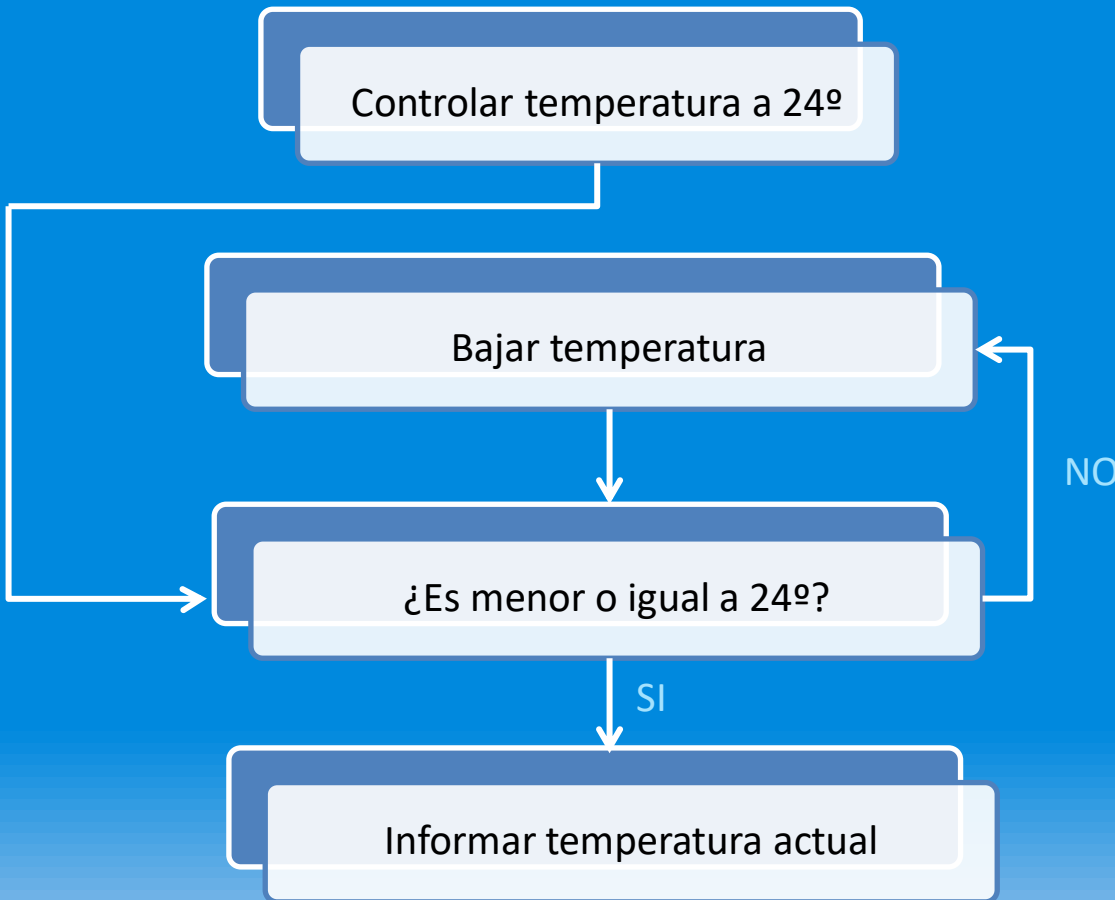
Adopta esquemas reiterativos de forma que es la repetición n veces de un método de búsqueda o iteración la que da lugar a unos resultados.



**Mientras** condición **Hacer**  
    secuencia\_de\_acciones  
**Fin Mientras**

## PROBLEMA CON ITERACIONES

Adopta esquemas reiterativos de forma que es la repetición n veces de un método de búsqueda o iteración la que da lugar a unos resultados.



**Mostrar** 'Ingresar temperatura'

**Leer** Temp

**Mientras** Temp > 24 **Hacer**

Temp <- Temp - 1

**Mostrar** 'Temp. Actual', Temp

**Fin Mientras**

**Mostrar** Temp

## EJERCITACIÓN

1. Asignar valores a 2 variables desde código y mostrar el contenido de la mayor por pantalla
2. Determinar si una persona es mayor o menor de acuerdo a su edad. Mayoría de edad a partir de los 21 años.
3. Leer 3 números y mostrar el mayor de ellos.
4. Leer 4 números y mostrar el máximo y el mínimo de ellos
5. Solicitar la contraseña 3 veces y mostrar el mensaje “contraseña correcta” o “contraseña incorrecta” según corresponda.
6. Solicitar números indefinidamente hasta que se ingrese un 0 (cero), luego mostrar el promedio de todos ellos



# **INSTITUTO TECNOLÓGICO BELTRÁN**

Centro de Tecnología e Innovación