



# ELEMENTOS DE COMPUTACIÓN Y LÓGICA



Universidad Nacional de Tucumán



2020 - AÑO DEL GENERAL MANUEL BELGRANO

## Contenido

**Unidad 1.** Propositiones Simples y Compuestas. Representación simbólica. Conectivos lógicos: Negación, Conjunción, Disyunción, Condicional, Bicondicional. Tablas de verdad. Jerarquía de conectivos. Clasificación de fórmulas lógicas por su significado: Tautologías, contradicciones y contingencias. Implicación y equivalencia lógica. Leyes Lógicas

**Unidad 2.** Razonamientos. Componentes: premisas, conclusión. Reglas de inferencia. Métodos de demostración: directo e indirecto. Validez de un razonamiento. Consistencia de Premisas.

**Unidad 3.** Lógica de Predicados. Funciones proposicionales. Universo del discurso. Representación simbólica. Cuantificadores: Cuantificador universal y cuantificador existencial. Alcance de un cuantificador. Variables libres y variables vinculadas. Propositiones categóricas. Negación de proposiciones cuantificadas. Equivalencia de proposiciones cuantificadas universalmente y existencialmente. Razonamientos. Reglas de especificación universal y existencial. Reglas de generalización universal y existencial.

**Unidad 4.** Álgebras Booleanas y circuitos combinatorios. Propiedades de los circuitos combinatorios. Funciones Booleanas.

**Unidad 5.** Sistemas de numeración binario. Conversiones. Suma y resta de binarios. Sistema de numeración octal. conversiones. Sistema de numeración hexadecimal.

**Unidad 6.** Diseño de Algoritmos. La programación como una metodología. Diseño de Algoritmos. Métodos de refinamientos sucesivos. Lenguaje de Diseño de programas. La programación estructurada. Estructuras algorítmicas fundamentales. Formas de reducción de complejidad: secuenciación, análisis por casos, análisis Iterativo. Generalización del concepto de procedimiento: Acciones parametrizadas, funciones.

**Unidad N° 6:** Diseño de Algoritmos. La programación como una metodología. Diseño de Algoritmos. Métodos de refinamientos sucesivos. Lenguaje de Diseño de programas. La programación estructurada. Estructuras algorítmicas fundamentales. Formas de reducción de complejidad: secuenciación, análisis por casos, **análisis Iterativo**.

## REPASEMOS LO VISTO

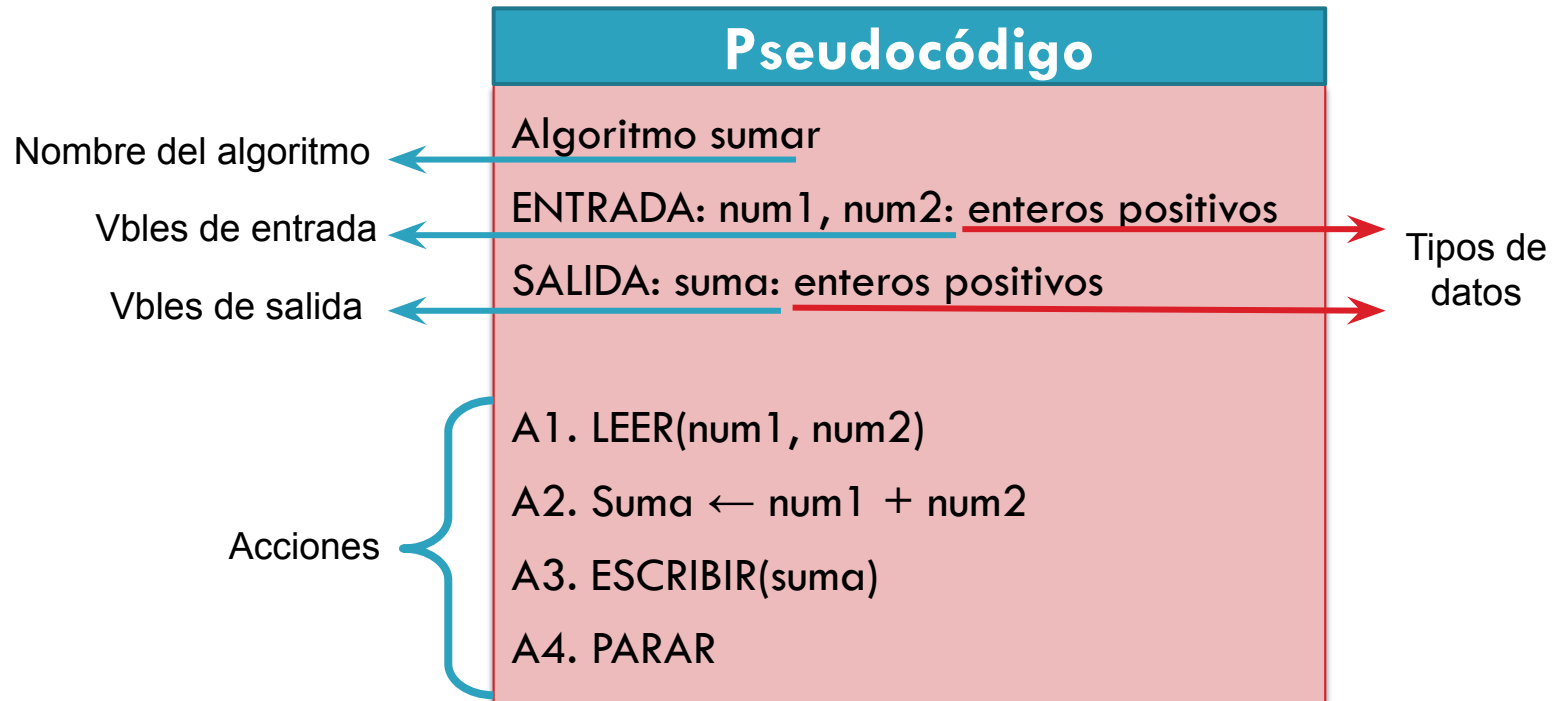
Un Algoritmo es una secuencia finita de pasos.

El diseño del algoritmo se realiza usando un PSEUDOCÓDIGO.

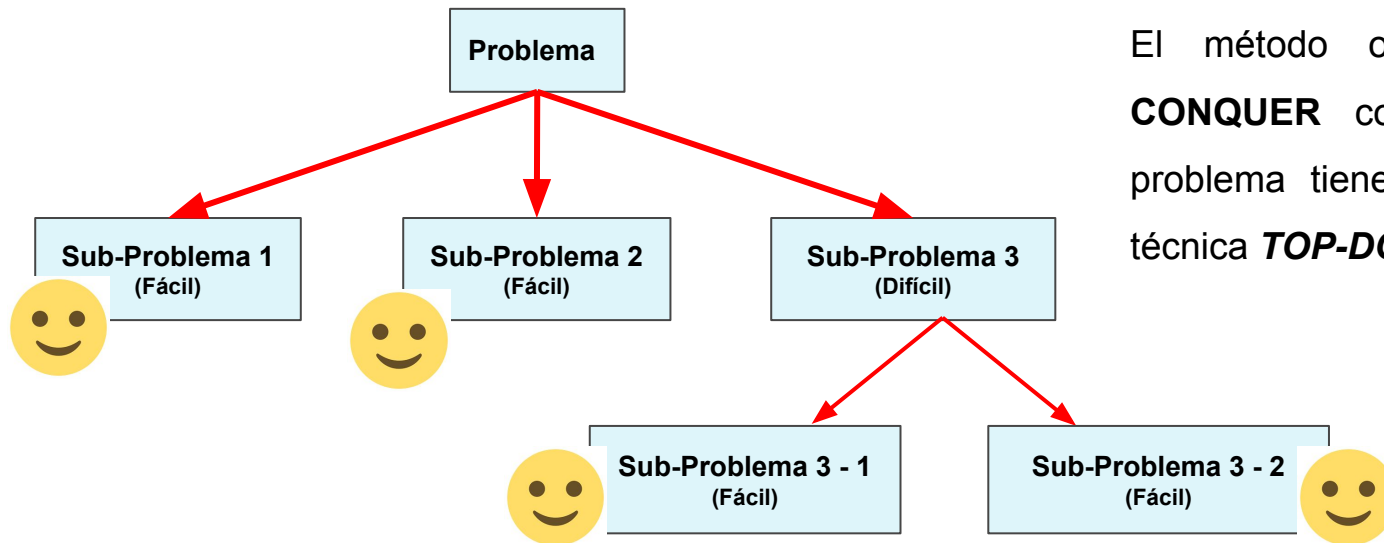
El algoritmo debe ser:

- ✓ **Neutro:** Es independiente al lenguaje que se vaya a usar.
- ✓ **Completo:** Permite expresar cualquier idea computacional.
- ✓ **Simple:** es decir fácil de entender y de escribir.
- ✓ **Eficiente:** Buen uso de los recursos de memoria y tiempo del proceso.

## REPASEMOS LO VISTO



## REPASEMOS LO VISTO



El método o técnica: **DIVIDE AND CONQUER** con la que se ataca un problema tiene la característica de ser una técnica **TOP-DOWN**.

Las **estructuras básicas** o fundamentales de la programación estructurada son:

- SECUENCIACIÓN
- SELECCIÓN
- ITERACIÓN

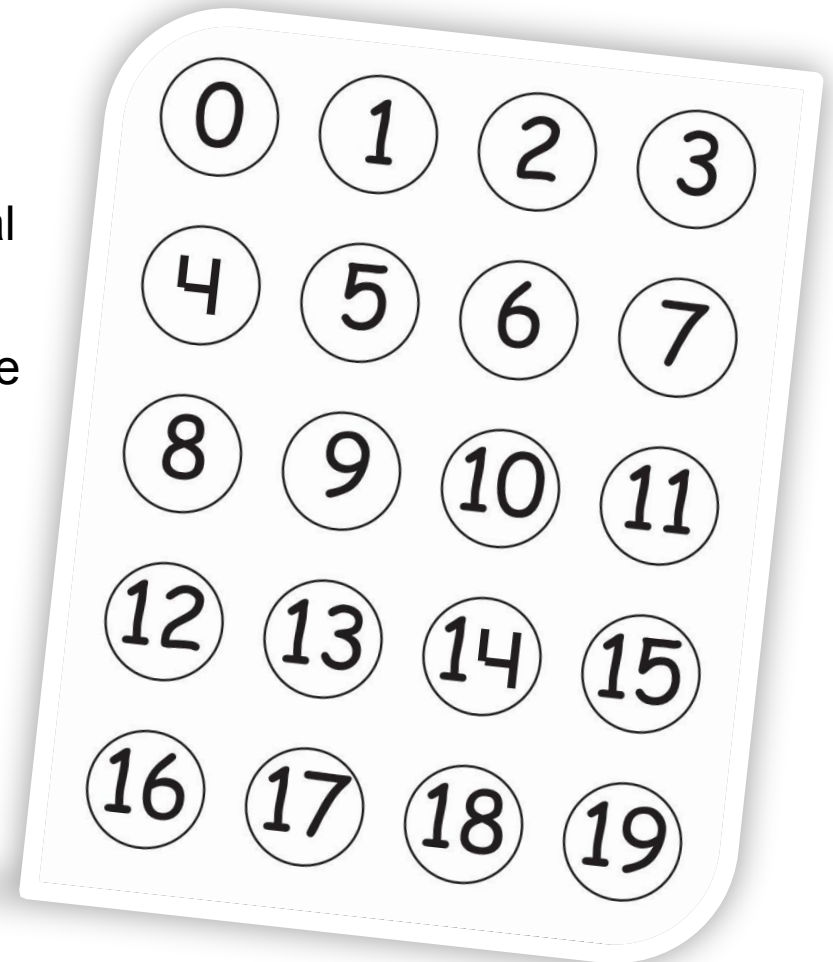
**SI** condición **ENTONCES**  
    A1  
**SINO**  
    A2

### SEGÚN expresión

∈ conjunto de valores1: A1;  
∈ conjunto de valores2: A2;  
∈ conjunto de valores3: A3;  
...  
**SINO** : An;

## ESTRUCTURAS DE SELECCIÓN

**Ejercicio:** realice un algoritmo, que al ingresar 2 números enteros los ordene de forma ascendente



# ESTRUCTURAS DE ITERACIÓN

Se utiliza cuando se debe aplicar un mismo tratamiento a un gran número de objetos. El número de estos objetos puede ser fijo o que no se conozca de antemano. Con esta estructura se reduce la complejidad del problema.

El tratamiento iterativo se aplica a las "secuencias de objetos".

## La secuencia de objetos:

- Puede tener un número fijo.
- Estar terminada por un elemento de características fácilmente distinguibles del resto de los objetos ☐ *"marca final"*.

### Iteración Condicional

(cantidad no conocida de iteraciones):

MIENTRAS condición

Acciones;

Fin\_Mientras

### Iteración de un número fijo de veces:

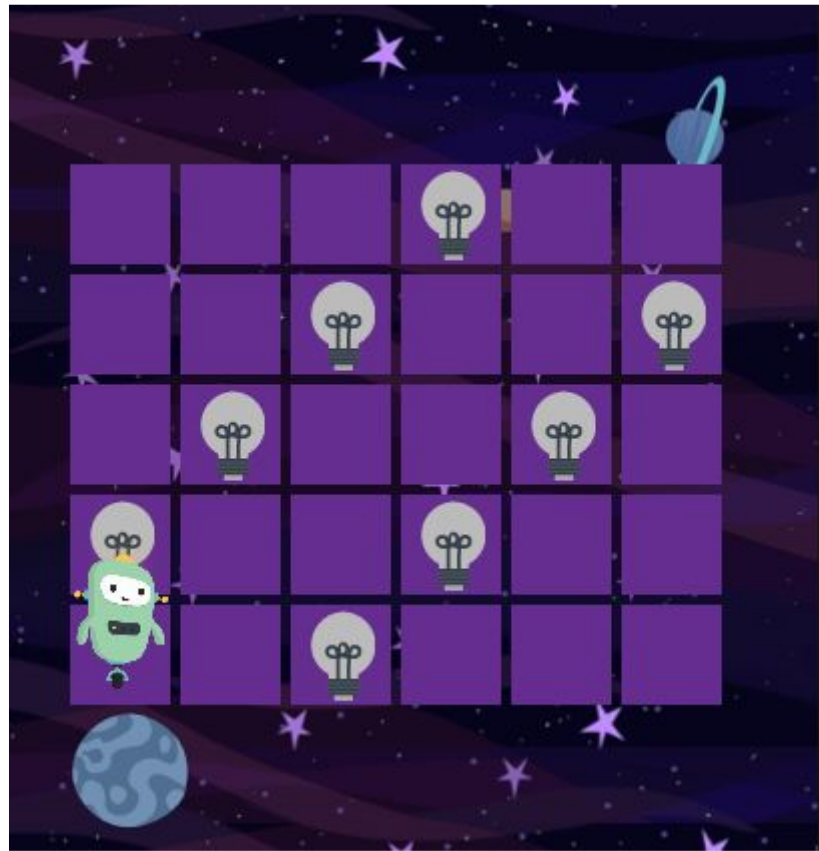
HACER n VECES

Acciones;

Fin\_Hacer

## VEAMOS UN EJEMPLO DE ITERACIÓN DE NÚMERO FIJO:

<https://pilasbloques.program.ar/online/#/desafio/5>





**Ejemplo:** Se desea leer una secuencia de 10 números enteros pero solamente escribir los que son múltiplos de 4, y contarlos.

**Algoritmo: Múltiplos de 4**

**ENTRADA:** num: entero

**SALIDA:** num: entero, cont: entero

**A0.** cont  $\leftarrow$  0

**A1.** HACER 10 VECES

    LEER (num)

    SI (RESTO (num, 4) = 0) ENTONCES

        cont  $\leftarrow$  cont+1

        ESCRIBIR (num)

    (fin del SI)

(fin del HACER)

**A2.** ESCRIBIR (cont) ;

**A3.** PARAR

**Ejemplo:** Se desea analizar los 10 primeros números enteros, escribir los que son múltiplos de 4 y contarlos.

**Algoritmo: Múltiplos de 4**

**ENTRADA:** -

**SALIDA:** num: entero, cont: entero

**Variable Auxiliar:** i : entero positivo

**A0.** cont  $\leftarrow$  0

**A1.** HACER 10 VECES (i: 1..10)

    SI (RESTO (i, 4) = 0) ENTONCES

        cont  $\leftarrow$  cont+1

        num  $\leftarrow$  i

        ESCRIBIR (num)

    (fin del SI)

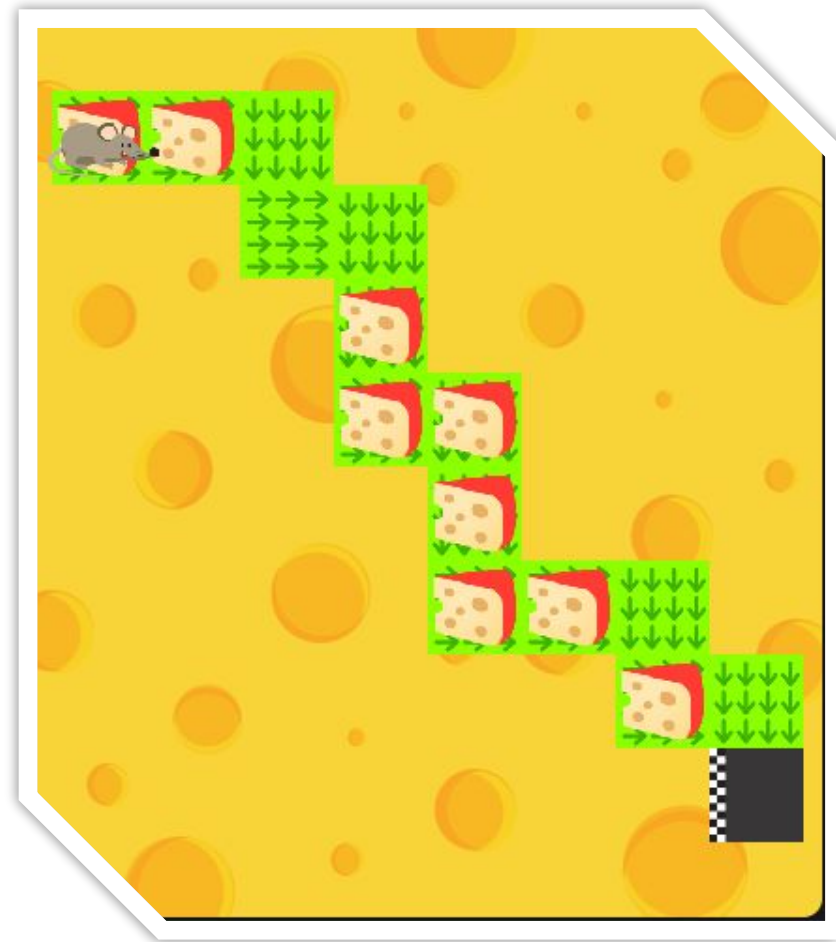
(fin del HACER)

**A2.** ESCRIBIR (cont);

**A3.** PARAR

## VEAMOS UN EJEMPLO DE ITERACIÓN CONDICIONAL

<https://pilasbloques.program.ar/online/#/desafio/21>



**Ejemplo:** Se desea leer una secuencia de caracteres con MF '.' y contar cuantos caracteres tiene en total.

**Algoritmo: Caracteres**

**Entrada:** cc: cadena de caracteres con MF='.'

**Salida:** cont: entero

```
A1. cont ← 0
A2. LEER(cc)
A3. MIENTRAS (cc <> MF)
    p3.1 cont ← cont + 1
    p3.2 LEER(cc)
    (fin del MIENTRAS)
A4. ESCRIBIR(cont)
A5. PARAR
```

## Ejercicios:

Escriba un algoritmo que permita leer el DNI y calcular la nota promedio de dos parciales de una asignatura, para 10 alumnos. Determine si cada estudiante promociona (promedio  $\geq 7$ ), regulariza (promedio  $\geq 4$ ) o recupera (uno o ambos parciales desaprobados).

Modifique el algoritmo anterior para leer la información de una cantidad desconocida de alumnos. ¿Qué estrategia utilizaría para detener la iteración condicional?

$\neg q$   
 $p \wedge p \leftrightarrow q$   
 $p \wedge \neg q$   
 $p \vee q$   
 $p \leftrightarrow q$   
 $\neg p$   
 $p \square q$   
 $\neg q$   
 $p \vee q$   
 $p \square q$   
 $\neg q$

Hasta la  
próxima clase

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F