


The logo for Silicon Misiones features the text "Silicon Misiones" in a bold, white, sans-serif font. The text is centered within a horizontal, rounded rectangular shape composed of several overlapping, semi-transparent colored areas: a red circle on the left, a light blue shape at the top, a green shape in the middle, a light green shape at the bottom, and a blue circle on the right.

# Silicon Misiones

Misiones, República Argentina

The page is decorated with large, abstract, curved shapes in various colors. A light green shape is in the top-left corner. In the bottom-right corner, there are three overlapping shapes: a teal one, a blue one, and a red one.

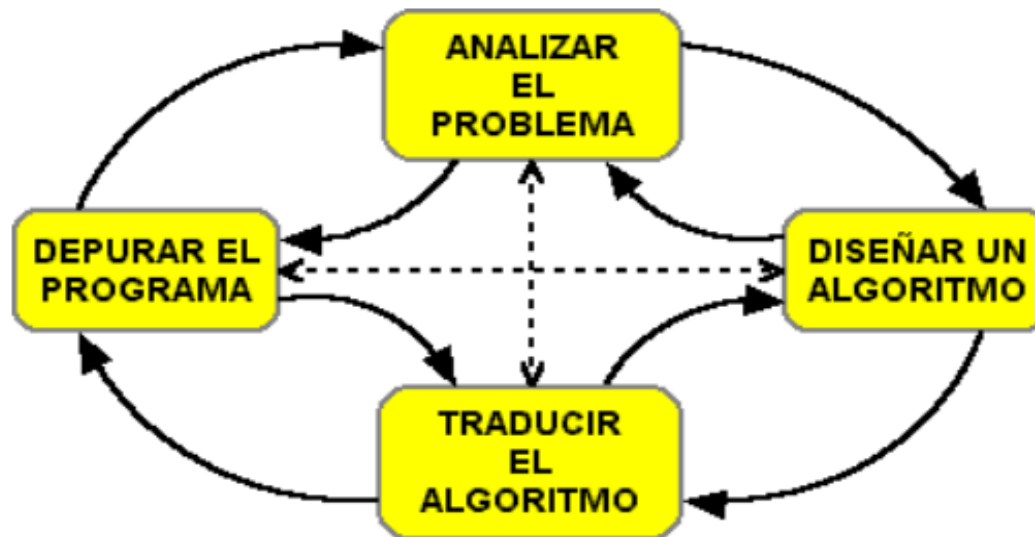
# **INTRODUCCIÓN A LA PROGRAMACIÓN**

## **CLASE 3**

# ALGORITMOS

- ❑ Un algoritmo es una lista de instrucciones bien definida, ordenada y finita mediante las cuales se efectuará paso a paso un proceso para obtener un cierto resultado, un ejemplo de ello son las recetas de cocina, manuales, etc
- ❑ Los programas de computadora tienen como finalidad resolver problemas específicos y el primer paso consiste en definir con precisión el problema hasta lograr la mejor comprensión posible.

Una forma de realizar esta actividad se basa en formular claramente el problema, especificar los resultados que se desean obtener, identificar la información disponible (datos), determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles (materia prima) en la información requerida (resultados).

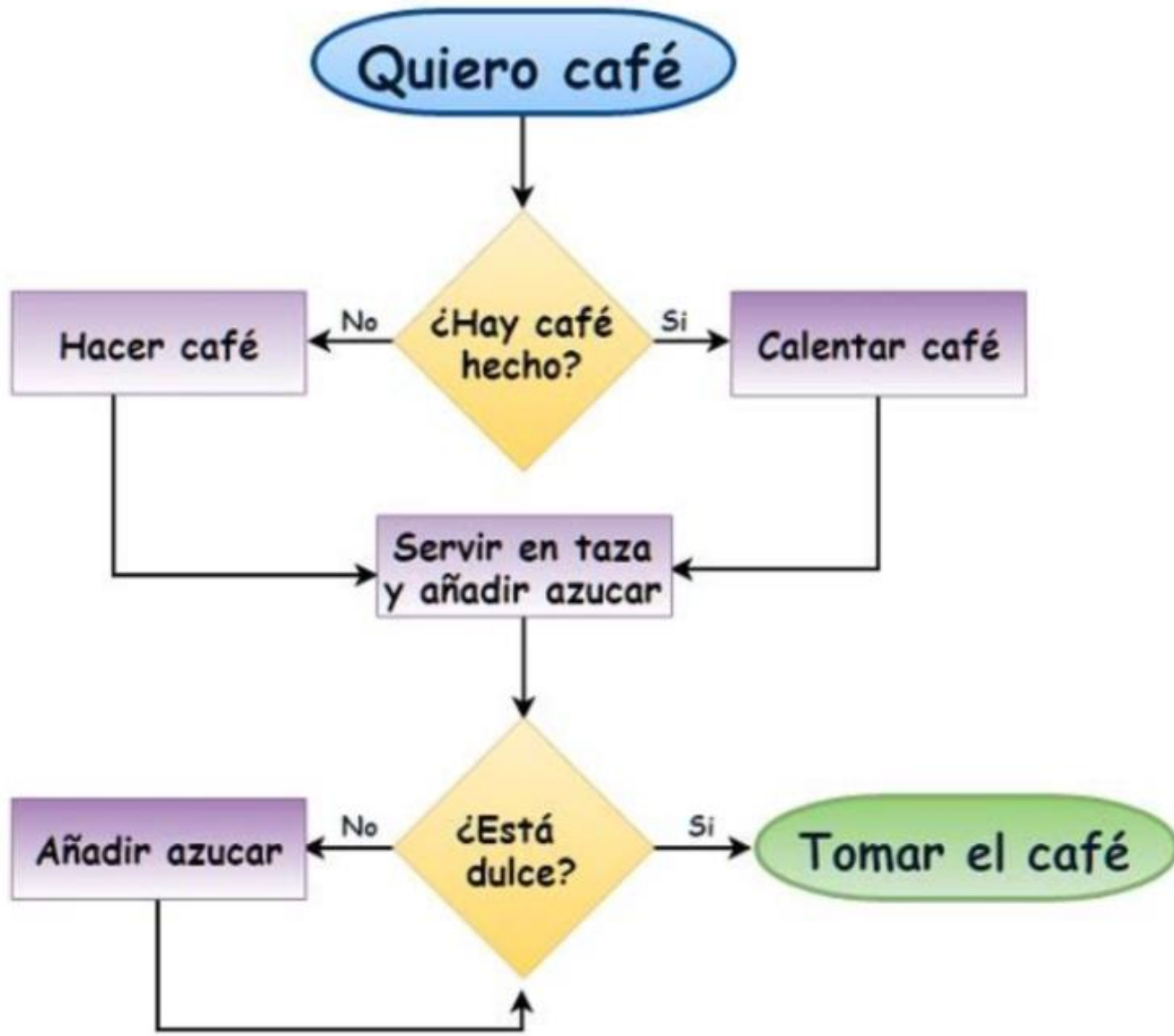


Fases para elaborar un programa

## DIAGRAMAS DE FLUJO

- Un diagrama de flujo o flujograma es una forma de representar gráficamente los pasos para resolver un problema en específico. Estos diagramas utilizan una serie de símbolos con significados especiales y son la representación gráfica de los pasos de un proceso. En computación es un primer enfoque con lo que sería la programación formal.





# DESARROLLO DE UN ALGORITMO

## 1. DEFINIR VARIABLES

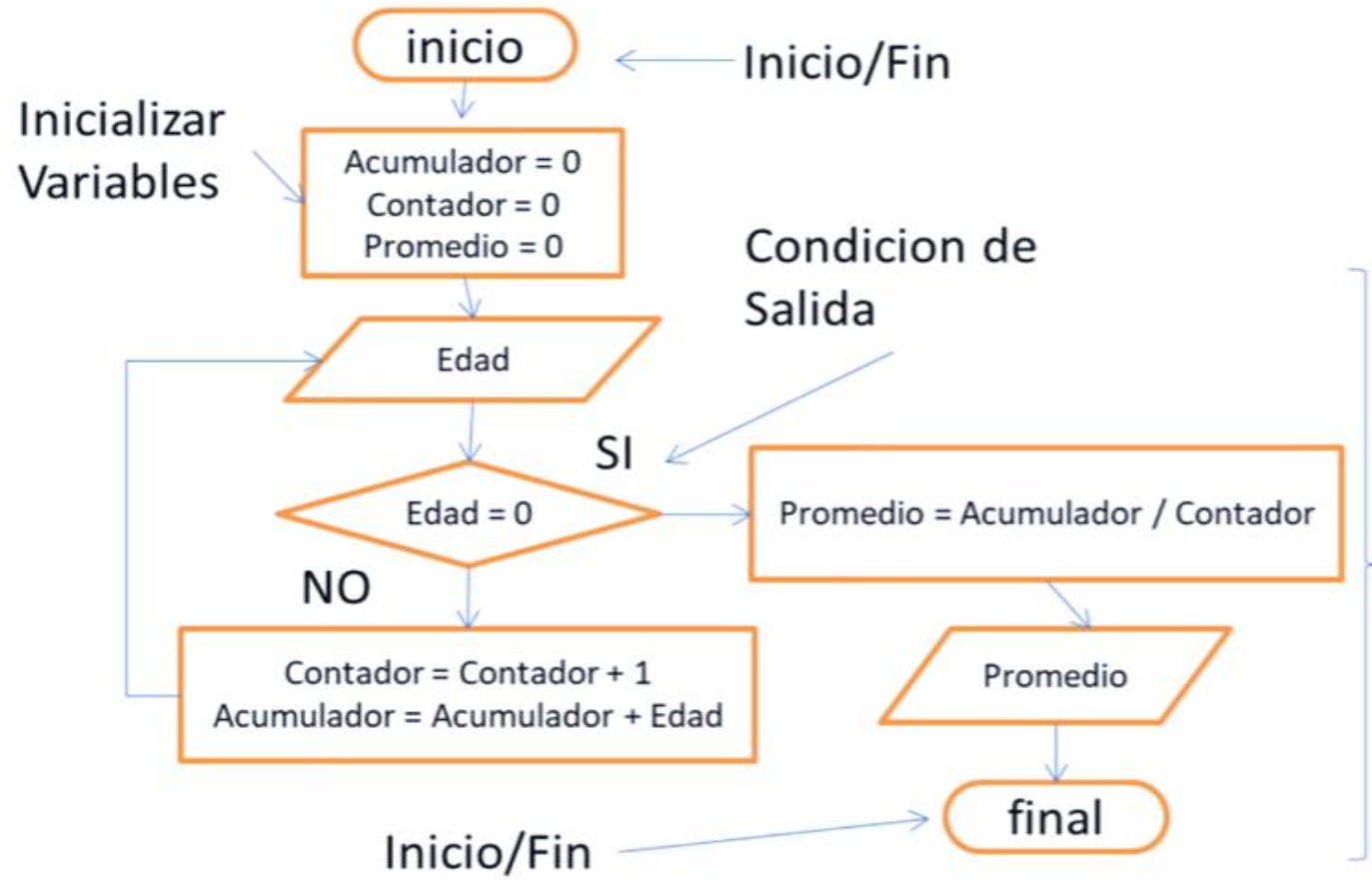
```
1  Algoritmo Tipos_De_Datos
2
3      definir nombre Como Caracter;
4      definir num Como Entero;
5      definir log Como Logico;
6      definir num_real como real;|
7
8
9  FinAlgoritmo
```

## 2. INICIALIZAR VARIABLES

## 3. DEFINIR CONDICIÓN DE SALIDA

## 4. RESOLUCIÓN DEL ALGORITMO

# DESARROLLO DE UN ALGORITMO





# OPERADORES LÓGICOS

- Los operadores son símbolos que se usan para comparar dos valores. Si el resultado de la comparación es correcto la expresión considerada es verdadera, caso contrario es falsa.

Operador	nombre	ejemplo	significado
<	menor que	$a < b$	$a$ es menor que $b$
>	mayor que	$a > b$	$a$ es mayor que $b$
==	igual a	$a == b$	$a$ es igual a $b$
!=	no igual a	$a != b$	$a$ no es igual a $b$
<=	menor que o igual a	$a <= 5$	$a$ es menor que o igual a $b$
>=	mayor que o igual a	$a >= b$	$a$ es menor que o igual a $b$

# ESTRUCTURAS CONDICIONALES

- ❑ Las estructuras condicionales comparan una variable contra otro(s) valor(es), y en base al resultado de esta comparación, se sigue un curso de acción dentro del programa. Cabe mencionar que la comparación se puede hacer contra otra variable o contra una constante, según se necesite. Existen tres tipos básicos, las simples, las dobles y las múltiples...



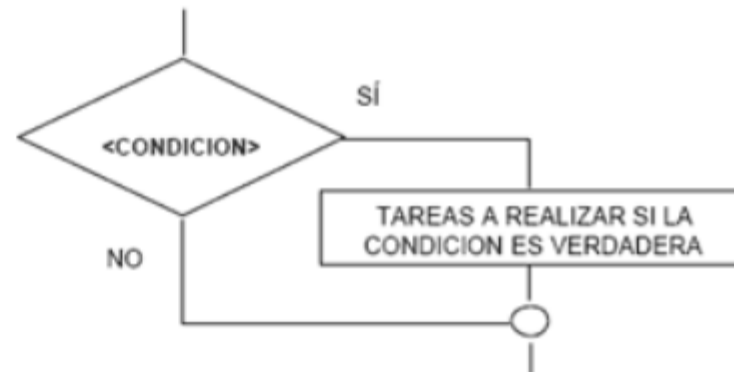
## ❖ SIMPLES

- Las estructuras condicionales simples se les conocen como “Tomas de decisión”. Estas tomas de decisión tienen la siguiente forma:

Pseudocódigo:

Si <condición> entonces  
Instrucción (es)  
Fin-Si

Diagrama de flujo:



## ❖ DOBLES

- Las estructuras condicionales dobles permiten elegir entre dos opciones o alternativas posibles en función del cumplimiento o no de una determinada condición. Se representa de la siguiente forma:

Pseudocódigo:  
Si <condición> entonces  
    Instrucción (es)  
Si no  
    Instrucción (es)  
Fin-Si



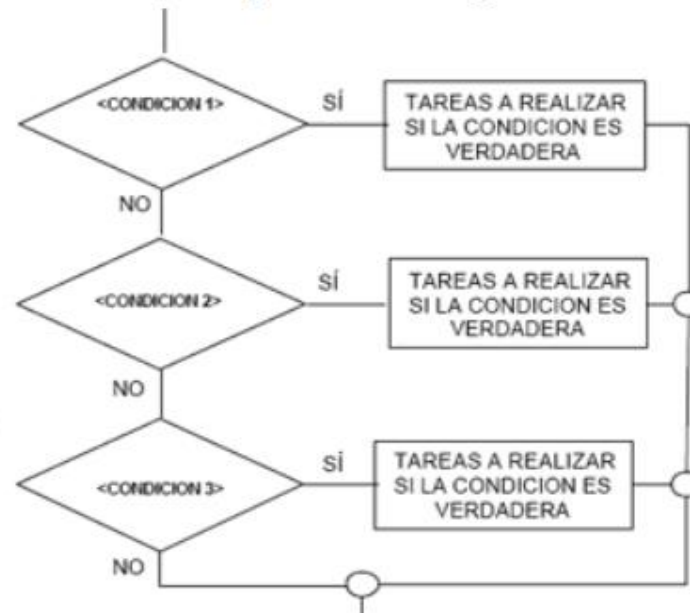
## ❖ MÚLTIPLES

- Las estructuras de comparación múltiples, son tomas de decisión especializadas que permiten comparar una variable contra distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas. La forma común es la siguiente:

Pseudocódigo:

```
Si <condición> entonces
  Instrucción(es)
Si no
  Si <condición> entonces
    Instrucción(es)
  Si no
    .
    . Varias condiciones
    .
```

Diagrama de flujo:



## ❖ ESTRUCTURA SEGÚN - HACER (CASOS)

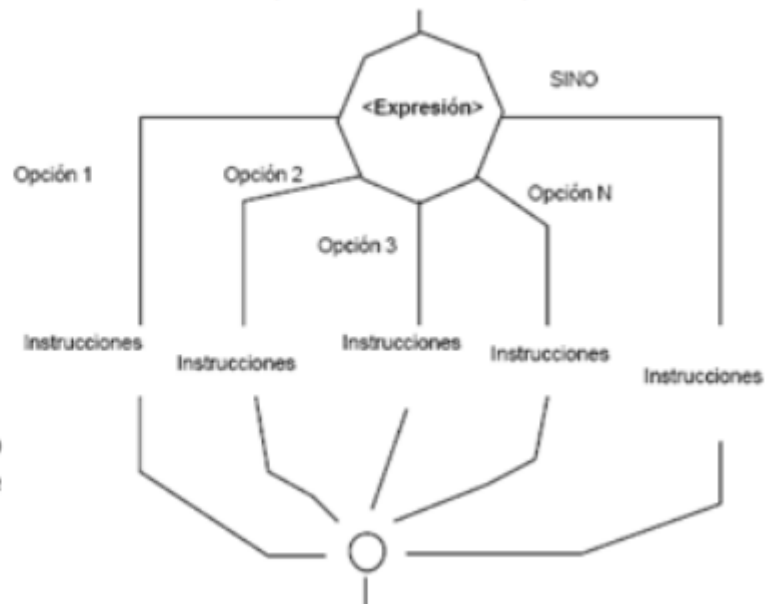
- Es una toma de decisión especializada que permite evaluar una variable con distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas. La forma es la siguiente:

Pseudocódigo:

```

En-caso-de <expresión> haga
  Caso <opción 1>:
    <instrucciones>
  caso <opción 2>:
    <instrucciones>
  caso <opción 3>:
    <instrucciones>
  ...
  caso <opción N>:
    <instrucciones>
  SINO <instrucciones a realizar si no
    se ha cumplido Ninguna de
    las condiciones anteriores>
Fin-Caso
  
```

Diagrama de flujo:



# EJERCICIOS QUE DEBEN SER PRESENTADOS

1. Hacer un algoritmo que permita que Dados dos números, indicar la suma, resta, división y multiplicación de ambos
2. Hacer un algoritmo que muestre por pantalla EL PROMEDIO de los primeros N números enteros.
3. Hacer un algoritmo que dada la nota final de un alumno imprima por pantalla su estado según:

Nota final < 6	«DESAPROBADO»
6 >= Nota final < 8	«APROBADO»
8 >= Nota final <= 9	«DISTINGUIDO»
Nota final = 10	«EXCELENTE»
4. Hacer un algoritmo que según el número ingresado por el usuario (entre 1 y 7) muestre por pantalla el día correspondiente en la semana:  
Ejemplo: ingresa 1 y corresponde al «LUNES», 2 corresponde a «MARTES», etc.

**MUCHAS GRACIAS!!**

**Sigamos con los ejercicios ...**