

CLASE DE REPASO

INGENIERA SILVIA PATRICIA BARDELLI

CONCEPTOS FUNDAMENTALES

Algoritmo:

Es una secuencia finita y repetible de pasos que describe el proceso a seguir para solucionar un problema dado.

CONCEPTOS FUNDAMENTALES

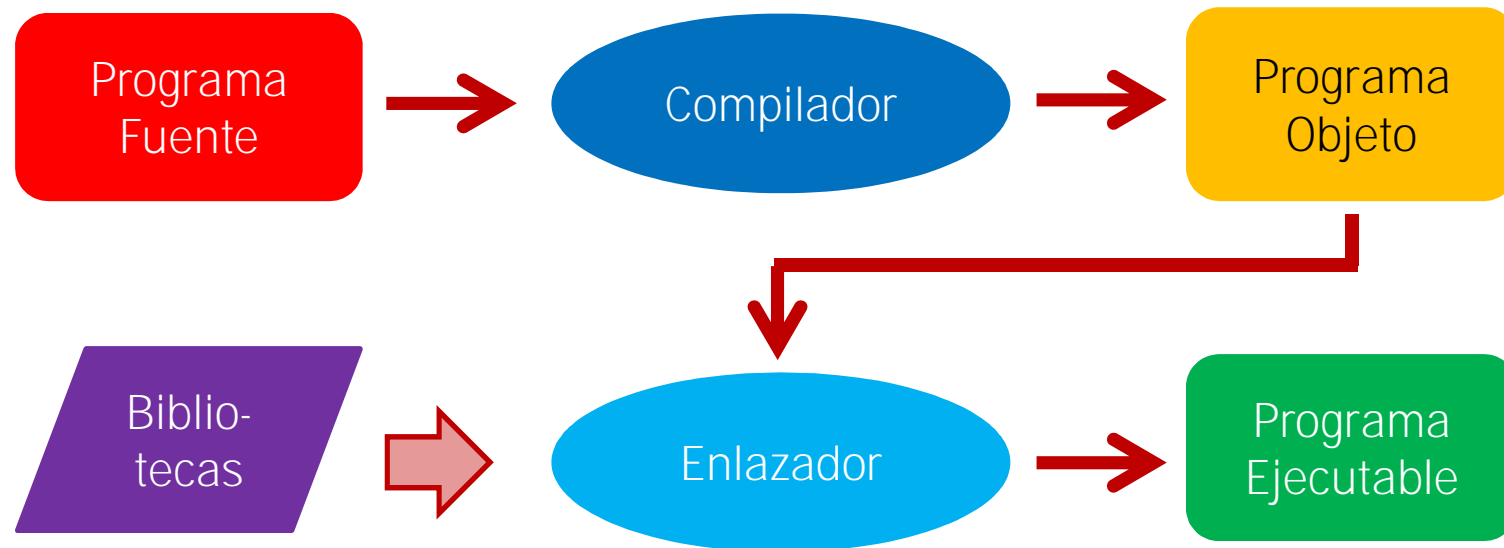
- § Secuencia: Significa que los pasos están ordenados.
- § Finita: Que tiene un final.
- § Repetible: Partiendo de las mismas condiciones iniciales, el resultado debe ser siempre el mismo.

CONCEPTOS FUNDAMENTALES

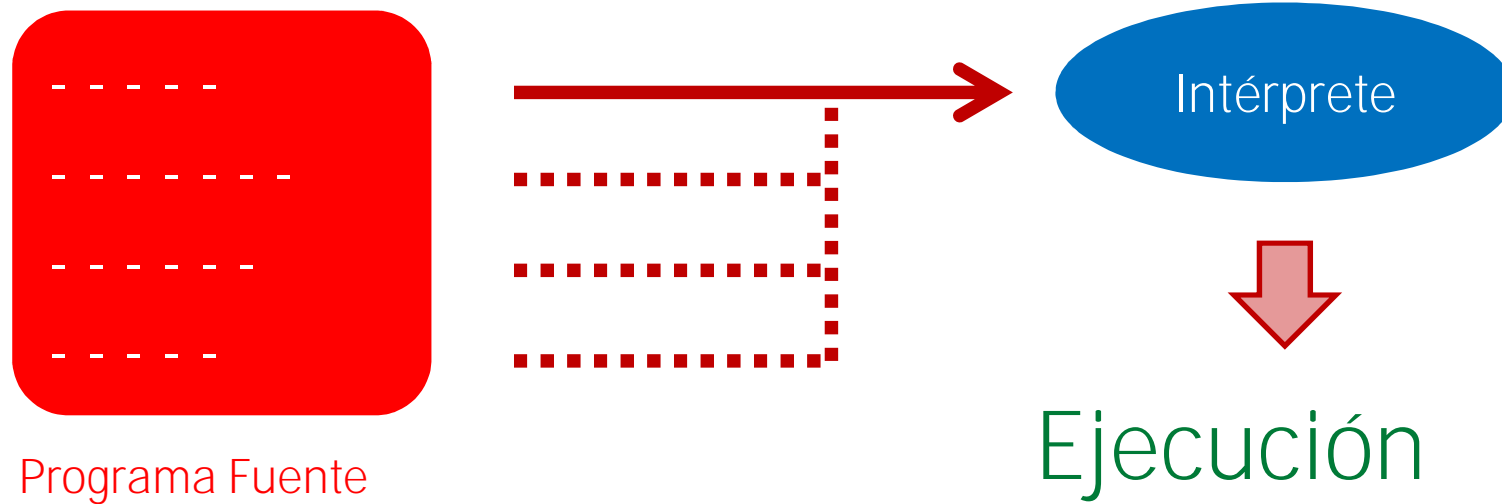
Programa:

Es la implementación de un algoritmo en algún lenguaje de programación.

PROCESO DE COMPILACIÓN



PROCESO DE INTERPRETACIÓN



COCINAR UNA HAMBURGUESA

- Encender el fuego de la hornalla.
- Colocar la plancha sobre él.
- Colocar la hamburguesa sobre la plancha o parrilla.
- 1. Esperar un momento
 - Determinar si está cocida del lado inferior.
 - Sí, continuar
 - No, ir a 1
 - Dar vuelta la hamburguesa
- 2. Esperar un momento
 - Determinar si está cocida del lado inferior
 - Sí, fin
 - No, ir a 2

Variables

Reglas para crear nombres de variables:

§Sólo letras, números y el guión bajo.

§No pueden comenzar con un número.

§No pueden coincidir con las palabras reservadas del lenguaje.

Variables

```
a = 3
```

Asignación de constante

```
b = a
```

Asignación de variable

```
c = a + b + 1
```

Asignación de expresión

```
pi = 3.1416
```

```
print("La variable a contiene", a, "b contiene", b, "y c  
contiene", c)
```

```
print("La variable pi contiene", pi)
```

Variables

+ Suma

* Multiplicación

// División
entera

** Potenciación

- Resta

/ División real

% Módulo o resto

Ingreso de datos por teclado

```
n = input("Ingrese un número entero: ")  
n = int(n)
```

-- - o también - - -

```
n = int(input("Ingrese un numero entero: "))
```

Además existe la función **float()** para poder ingresar números reales.

Ejemplo

Leer dos números enteros y guardarlos en dos variables.

Luego intercambiar sus valores e imprimir su contenido.

```
a = int(input("Ingrese un número entero: "))
```

```
b = int(input("Ingrese otro número entero: "))
```

```
print("A contiene", a, "y B contiene", b)
```

```
c = a
```

```
a = b
```

```
b = c
```

```
print("Ahora A contiene", a, "y B contiene", b)
```

Estructura Secuencial

- § Trabajando de esta manera las posibilidades de resolución de problemas son limitadas. Sólo cálculos e impresiones.
- § Para que un programa sea *realmente útil* es necesario que sea capaz de *tomar decisiones* y actuar en consecuencia.

Estructura Alternativa o Condicional

- § Por eso, además de la estructura secuencial, existen dos estructuras más en el mundo de la Programación Estructurada.
- § La segunda que veremos se denomina *Estructura Alternativa o Condicional*.

Instrucción if

Formato 1

if *<condición>*:

.

.

.

Ejemplo Nro 1

```
# Leer un número entero e imprimir un  
# mensaje indicando si es mayor que 5.  
n = int(input("Ingrese un número: "))  
if n > 5:  
    print("El número es mayor que 5")  
# Fin del programa
```


Instrucción if

Formato 2

if *<condición>*:

.

.

else:

.

.

Ejemplo Nro 2

*# Leer la calificación que obtuvo un alumno en un
examen final e imprimir un mensaje indicando si
aprobó o no la materia. Se aprueba con 4.*

```
nota = int(input("Ingrese la calificación: "))
```

```
if nota >= 4:
```

```
    print("El alumno aprobó la materia")
```

```
else:
```

```
    print("El alumno no aprobó la materia")
```

Instrucción if

Formato 3

if *<condición>*:

.

elif *<condición>*:

.

else:

.

Ejemplo Nro 3

Leer un número e informar si es positivo, negativo o cero.

```
n = int(input("Ingrese un número entero: "))
```

```
if n > 0:
```

```
    print("El número es positivo")
```

```
elif n < 0:
```

```
    print("El número es negativo")
```

```
else:
```

```
    print("El número es cero")
```

Operadores Lógicos

Operador *and* (*Y*):

Cond. 1	Cond. 2	Cond. 1 <i>and</i> Cond. 2
V	V	V
V	F	F
F	V	F
F	F	F

Operadores Lógicos

Operador *or* (*O*):

Cond. 1	Cond. 2	Cond. 1 <i>or</i> Cond. 2
V	V	V
V	F	V
F	V	V
F	F	F

Operadores Lógicos

Operador *not* (*NO*):

Condición	<i>not</i> Condición
V	F
F	V

Ejemplo Nro 4

Leer un número entero e imprimir un mensaje indicando

si corresponde a un número válido de mes.

```
mes = int(input("Ingrese un número de mes: "))
```

```
if mes >= 1 and mes <= 12:
```

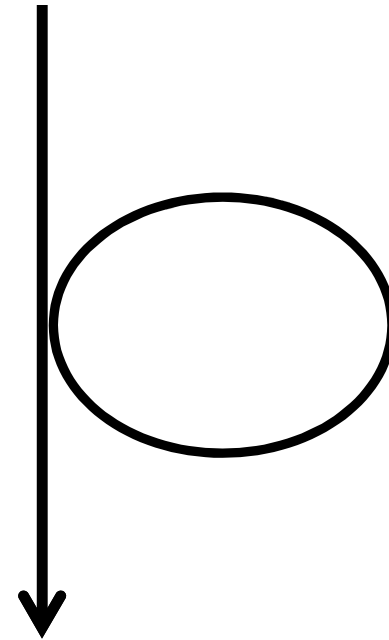
```
    print("El mes es válido")
```

```
else:
```

```
    print("El mes es inválido")
```


ESTRUCTURA ITERATIVA, CICLO O BUCLE

En esta estructura el programa repetirá una porción de su código una cierta cantidad de veces, y luego seguirá adelante.



INSTRUCCIÓN WHILE

while *<condición>*:

• • • • •

• • • • •

• • • • •

EJEMPLO

Imprimir los números enteros entre 1 y 100

a = 1

while a <= 100:

 print(a)

 a = a + 1

Fin del programa

DEFINICION CONTADOR

Cuando una variable es modificada en una cantidad **fija** respecto de su valor anterior, se la denomina ***contador***.

ATENCIÓN

¿Qué ocurre si olvidamos la línea que incrementa el contador?

```
a = 1
```

```
while a <= 100:
```

```
    print(a)
```

```
    a = a + 1
```

```
# Fin del programa
```

EJEMPLO

```
suma = 0
cant = 0
n = int(input("Ingrese un número o -1 para terminar: "))
while n != -1:
    suma = suma + n
    cant = cant + 1
    n = int(input("Ingrese un número o -1 para terminar: "))
if cant != 0:
    prom = suma/cant
    print("El promedio es", prom)
else:
    print("No se ingresaron valores")
```

DEFINICIÓN ACUMULADOR

Cuando una variable es modificada en una cantidad **cambiante** respecto de su valor anterior, se la denomina ***acumulador***.

EJEMPLO

Objetivo:

Leer un conjunto de números enteros e imprimir el mayor. El fin de los datos se indica con -1.

¿Cómo podemos proceder para hallar el máximo?

EJEMPLO

```
n = int(input("Ingrese un número o -1 para terminar: "))
mayor = n
while n != -1:
    if n > mayor:
        mayor = n
    n = int(input("Ingrese un número o -1 para terminar: "))
print("El mayor es", mayor)
```

Fin de la 1ra parte de la materia

PROFESORA ING. SILVIA PATRICIA BARDELLI