



ALGORITMIA Y PROGRAMACIÓN

ESTRUCTURAS ITERATIVAS

Contenido

- ❑ Estructuras de Repetición
 - Mientras que - While
 - Ejercicios
 - Contadores y acumuladores

Estructuras de Repetición

- ❑ **Ejercicio:** Suponga que debe mostrar los números del 1 a 100 en una aplicación.

Estructuras de Repetición

Ejercicio: Suponga que debe mostrar los números del 1 a 100.

Se podría hacer con la siguiente función:

```
Inicio mostrarNumeros()  
    imprimir("El número es: 1")  
    imprimir("El número es: 2")  
    imprimir("El número es: 3")  
    imprimir("El número es: 4")  
    imprimir("El número es: 5")  
    imprimir("El número es: 6")  
    imprimir("El número es: 7")  
    imprimir("El número es: 8")  
    imprimir("El número es: 9")  
    imprimir("El número es: 10").  
    ...  
    imprimir("El número es: 100")  
Fin
```



```
def mostrarNumeros():  
    print("El número es: 1")  
    print("El número es: 2")  
    print("El número es: 3")  
    print("El número es: 4")  
    print("El número es: 5")  
    print("El número es: 6")  
    print("El número es: 7")  
    print("El número es: 8")  
    ...  
    print("El número es: 100")
```

Y si nos piden listar del 1 al 5000?

Estructuras de Repetición

Son un grupo de instrucciones que permite la ejecución repetitiva de otro grupo de instrucciones. Hay una variable asociada al **ciclo** o **estructura de repetición** que controla el número de veces que se repetirán las instrucciones.

Existen 2 estructuras de repetición:

Mientras que → **while**

Para → **for**

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

1. Análisis del problema

Entradas: ??

Salidas: ??

Proceso:??

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

1. Análisis del problema

Entradas: n

Salidas: ??

Proceso:??

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

1. Análisis del problema

Entradas: n

Salidas:

- imprimir (número 1)
- imprimir (número 2)
- .
- .
- imprimir (número n)

Proceso:??

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

1. Análisis del problema

Entradas: n

Salidas:

imprimir (número 1)

imprimir (número 2)

.

.

imprimir (número n)

Proceso:

desde 1 hasta n

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

1. Análisis del problema

Entradas: n

Salidas:

imprimir (número 1)

imprimir (número 2)

.

.

imprimir (número n)

Proceso:

desde 1 hasta n

Imprimir 1,2,3,..... n

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

1. Análisis del problema

Entradas: n

Salidas:

imprimir (número 1)

imprimir (número 2)

...

imprimir (número n)

Proceso:

Mientras sea menor que n Haga

imprimir 1,2,3...n

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

2. Diseñar el algoritmo y escribirlo en pseudocódigo

Inicio

n: entero

leer (n)

???

Fin

Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

2. Diseñar el algoritmo y escribirlo en pseudocódigo

Inicio

n: entero

i: entero ← (Variable de control)

leer (n)

i = 1

Mientras $i \leq n$ Haga

Imprimir (i)

incrementar i en 1

Fin mientras

Fin

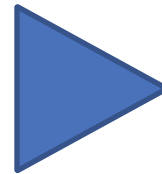
Estructuras de Repetición

Problema: Desarrollar un programa que solicite un número y muestre los números desde el 1 hasta el número solicitado.

3. Codificar el algoritmo usando algún lenguaje de programación

pseudocódigo

```
Mientras  $i \leq n$  Haga  
    Imprimir (i)  
    incrementar i en 1  
Fin mientras
```



Python



Estructura while

- ❑ Evalúa una condición dada para determinar si repite un conjunto de instrucciones. Si la condición es verdadera se repiten las instrucciones.
- ❑ Observe que el primer paso es validar la condición y luego se ejecutan las instrucciones (en caso de que la condición cumpla).

Pseudocódigo	Python
Mientras condición Haga instrucción 1 instrucción 2 Fin mientras	while (condición): instrucción 1 instrucción 2

Estructura Mientras que

Debe existir una variable de control del ciclo

La variable de control se inicializa antes del ciclo

Inicio

n: entero
i: entero

Leer (n)

i = 1

Mientras $i \leq n$ Haga

Imprimir (i)

incrementar i en 1

Fin mientras

Fin

La variable de control se compara contra el valor final

La variable debe actualizarse dentro de cada ciclo

Estructura Mientras que

Debe existir una variable de control del ciclo

La variable de control se inicializa antes del ciclo

La variable de control se compara contra el valor final

```
n = int(input("Digite un número: "))  
i = 0
```

```
while(i <= n):  
    print("El numero es: ", i)  
    i = i + 1
```

La variable debe actualizarse dentro de cada ciclo

Estructura while

❑ Ejemplo:

```
a=5
b=7
c=3
while (a < b):
    c = c *2
    a = a + 1

print (c)
print(a)
```

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

 c = c * 2

 a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

 c = c * 2

 a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7 verdadero
1				

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

c = c * 2

a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7 verdadero
1			6	

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

c = c * 2

a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7 verdadero
1	6		6	

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

c = c * 2

a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7 verdadero
1	6		6	6<7 verdadero

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

c = c * 2

a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7 verdadero
1	6		6	6<7 verdadero
2	7		12	

Estructura while

❑ Ejemplo:

a=5

b=7

c=3

while (a < b):

c = c * 2

a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	5	7	3	5<7 verdadero
1	6		6	6<7 verdadero
2	7		12	7<7 falso
Se interrumpe				

Estructura while

❑ Ejemplo:

a=9

b=7

c=3

while (a < b):

c = c * 2

a = a + 1

Número Iteración	Valor variable a	Valor variable b	Valor variable c	a<b
0	9	7	3	9<7
Se interrumpe				

Si la primera vez la condición del ciclo while no se cumple, **NO** se ejecuta ninguna instrucción.

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia y programación se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%)

Calcular la nota definitiva para un grupo de n estudiantes.

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia y programación se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%)
- ❑ Inicialmente calculamos la nota para un estudiante, conocidas sus 4 calificaciones:
- ❑ Análisis:
 - Entradas
 - Salida
 - Proceso

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%)
- ❑ Inicialmente calculamos la nota para un estudiante, conocidas sus 4 calificaciones:

```
calculaDefinitiva (p1, p2, lab, tra : real)
```

```
Inicio
```

```
    def = 0
```

```
    def = p1*0.3+p2*0.35+ lab*0.25+tra*0.1
```

```
    retornar def
```

```
Fin
```

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%)
- ❑ Prueba de escritorio del algoritmo:

calculaDefinitiva (p1, p2, lab, tra : real)

Inicio

def = 0

def = p1*0.3+p2*0.35+ lab*0.25+tra*0.1

retornar def

Fin

p1	p2	lab	tra
3,2	2,4	4,4	3,0
	def = 3,2		

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%). Hallar la nota definitiva para n estudiantes.
- ❑ ¿Qué tenemos que hacer?

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%). Hallar la nota definitiva para n estudiantes.
- ❑ Ahora trabajamos en el algoritmo principal:

Inicio

nombre: **texto**

par1, par2, lab, tra: **real**

numeroEst, j = 1: **entero**

Leer (numeroEst)

mientras (j <= numeroEst) **haga**

leer (nomb)

leer (par1)

leer (par2)

leer (lab)

leer(tra)

 definitiva = **calculaDefinitiva**(par1, par2, lab, tra)

Imprimir("La nota definitiva de ", nombre, " es: ", definitiva)

 j++

Fin mientras

Fin

```
Inicio calculaDefinitiva(p1, p2, l, t)
    d = p1*0.3 + p2*0.35 + l*0.25 + t*0.1
    retornar (d)
Fin-calculaDefinitiva
```


Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia se califica con dos parciales (el primero pesa 30%, el segundo 35%), laboratorios (25%) y trabajo final (10%). Hallar la nota definitiva para n estudiantes.
- ❑ Ahora elaboramos el programa principal a partir del algoritmo, en Python :

```
def calcularDefinitiva(p1, p2, l, t):  
    d = p1*0.3 + p2*0.35 + l*0.25 + t*0.1  
    return (d)  
  
#Programa principal  
numeroEst = int(input("Ingrese el número de estudiantes: "))  
j = 1  
while(j <= numeroEst):  
    nombre = input("Ingrese el nombre: ")  
    par1 = float(input("parcial 1: "))  
    par2 = float(input("parcial 2: "))  
    lab = float(input("Laboratorios: "))  
    tra = float(input("Trabajos: "))  
  
    definitiva = calcularDefinitiva(par1, par2, lab, tra)  
    print(definitiva)  
    print("la nota definitiva de ", nombre, " es: ", definitiva)  
    j+= 1
```

Estructura while - Contadores

Son variables usadas para realizar, como su nombre lo indica, conteos de la cantidad de veces que se cumple una situación específica.

Como su objetivo principal es contar, deben ser de tipo entero y normalmente se inicializan en cero.

Los contadores pueden ser usados en cualquier tipo de ciclos.

Estructura while - Contadores

Diseñe un algoritmo que permita contar los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

1 2 3 4 5 6 7 8 n

Estructura while - Contadores

Diseñe un algoritmo que permita contar los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Entrada: ?

Salida: ?

Proceso: ?

Estructura while - Contadores

Diseñe un algoritmo que permita contar los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Entrada: n

Salida: cantidad de números pares entre
1 y n

Proceso:

Estructura while - Contadores

Diseñe un algoritmo que permita contar los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Entrada: n
Salida: cantidad de números pares entre 1 y n
Proceso:

```
i = 1
cp = 0
mientras (i <= n) haga
    si ((i % 2) == 0)
        cp += 1
    fin-si
    i += 1
fin-mientras
```

Estructura while - Contadores

Diseñe un algoritmo que permita contar los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Algoritmo:

Inicio

n: entero

Leer (n)

i = 1

cp = 0

mientras (i <= n) haga

si ((i%2) == 0) entonces

cp += 1

fin-si

i += 1

Fin mientras

imprimir (cp)

Fin

Inicializa el contador

incrementa el contador

Estructura while - Contadores

Diseñe un algoritmo que permita contar los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Algoritmo:

Inicio

n: entero

Leer (n)

i = 1

cp = 0

mientras (i <= n) haga

si ((i%2) == 0) entonces

cp += 1

fin-si

i += 1

Fin mientras

imprimir (cp)

Fin

```
n = int(input("Ingrese un numero: "))
i = 1
cp = 0
while(i <= n):
    if(i%2 == 0):
        cp += 1

    i += 1

print(cp)
```


Estructura While-Acumuladores

Son variables usadas para ir acumulando el resultado de una operación.

Pueden ser de tipo numérico (entero o real) en las cuales acumula el resultado de operaciones matemáticas, o de tipo cadena en las cuales se concatenan frases o palabras.

Estructura While - Acumuladores

Diseñe un algoritmo que permita contar y **sumar** los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Inicio

n: entero

Leer (n)

i = 1

cp = 0

suma = 0

mientras (i <= n) haga

si ((i%2) == 0) entonces

cp += 1

suma = suma + i

fin-si

i += 1

Fin mientras

imprimir (cp, suma)

Fin

La variable **suma** va acumulando el valor de la suma de los números pares en cada iteración del ciclo principal.



Estructura While - Acumuladores

Diseñe un algoritmo que permita contar y **sumar** los números pares existentes en una serie de 1 a n, siendo n un número digitado por un usuario.

Inicio

n: entero

leer (n)

i = 1

cp = 0

suma = 0

mientras (i <= n) haga

 si ((i%2) == 0) entonces

 cp += 1

 suma = suma + i

 fin-si

 i += 1

Fin mientras

imprimir (cp, suma)

Fin

```
n = int(input("Ingrese un numero: "))
i = 1
cp = 0
suma = 0
while(i <= n):
    if(i%2 == 0):
        cp += 1
        suma = suma + i

    i += 1

print(cp, suma)
```

Ejercicio Estructura while

- ❑ **Ejercicio:** Para ingresar al curso de Producción Cinematográfica se realizó un prueba clasificatoria. Se tienen los resultados de dicho examen por aspirante (una nota comprendida entre 0.0 y 5.0).

Se desea saber cuántos aspirantes aprobaron el examen, cuántos lo perdieron (nota menor que 3.0) y cuál fue el promedio de todo el grupo de aspirantes. No sabemos cuántos aspirantes son, pero sabemos que cuando se quiera indicar que se finalizó el ingreso de notas se digitará un valor negativo.

Ejercicio Estructura while

determinarEstadisticasSegunNotas()

Inicio

nota, promedioCalificaciones: **real** numAprobados, numReprobados : **entero**

leer(nota)

Mientras **nota** >= 0 **Haga**

Si (nota <= 5)

Si (nota >= 3)

 numAprobados = numAprobados + 1

sino

 numReprobados = numReprobados + 1

 promedioCalificaciones = promedioCalificaciones + nota

leer(nota)

Fin mientras

promedioCalificaciones = promedioCalificaciones / (numAprobados + numReprobados)

imprimir(numAprobados, numReprobados, promedioCalificaciones)

Fin

Ejercicio Estructura while

```
def determinarEstadisticasSegunNotas():
    promedioCalificaciones = 0
    numAprobados = 0
    numReprobados = 0
    nota = float(input("Digite la nota: "))
    while(nota>=0):
        if(nota<=5):
            if(nota>=3):
                numAprobados += 1
            else:
                numReprobados += 1
            promedioCalificaciones +=nota
        nota = float(input("Digite la nota: "))
    promedioCalificaciones /= (numAprobados+numReprobados)
    print("Número Aprobados: ",numAprobados)
    print("Número Reprobados: ",numReprobados)
    print("Promedio Notas: ",promedioCalificaciones)
```

Ejercicios Estructura while

1. Diseñe un algoritmo que reciba números digitados por el usuario y los imprima en pantalla, el programa debe terminar cuando el usuario digite un número negativo.
2. Diseñe un algoritmo que reciba números digitados por el usuario y al final imprima la suma de los números digitados, el programa debe terminar cuando el usuario digite 0 (cero).
3. Cree un algoritmo que permita digitar palabras y al final imprima la concatenación de las palabras digitadas, el programa debe terminar cuando el usuario digite la palabra “terminar”.

NOTA:

Para cada ejercicio realice su respectiva implementación en Python

Ejercicios Estructura while

4. Suponga que el administrador de una tienda desea calcular el total de ventas del día, para ello debe registrar el nombre del producto, el valor del producto y la cantidad de unidades vendidas. Al final el sistema debe mostrar la lista de productos digitados, el total de ventas por cada producto, el total de ventas del día y el valor global cobrado por concepto del IVA.

El administrador puede registrar productos hasta que él decida que ha terminado para lo cual digita un dato negativo.

NOTA: Realice la respectiva implementación en Python.

Ejercicios Estructura while

5. Suponga que el cálculo de la pensión de una persona se realiza de la siguiente manera: por cada año de servicio se paga \$80 si el empleado ingresó en o después de 1995 y \$100 si ingresó antes, dicho valor (80 o 100) se multiplica por el número de cada año más la edad que tenía en el año (ej $(100 \cdot 1994 + 32) + (100 \cdot 1995 + 33) + \dots$), el descuento de seguridad social en salud es del 12%. El programa debe recibir el año de ingreso y la edad del empleado en el año de ingreso, devolver el sueldo o mesada bruta, la mesada neta y el valor del descuento por salud.

Ejemplo: Para una persona que ingresó en el 2009 y que tenía 44 años en dicho año, su mesada o sueldo bruto para el 2011 es \$482.535, el descuento por salud es \$57.904 y por lo tanto su sueldo o mesada neta es \$424.630.

NOTA: Realice la respectiva implementación en Python.

Ejercicio Estructura while

- ❑ **Ejercicio:** El curso de Algoritmia se califica con dos parciales (el primero tiene un peso de 30% y el segundo 35%), una nota de laboratorios (25%) y una nota del trabajo final del curso (10%).
- ❑ Se pide calcular la nota definitiva, pero ahora NO se conoce el número de estudiantes. Cómo sabría cuando parar?