



Docente: Mg. Bryan Fernández Ebrath.

Practica. Estructuras repetitivas.

Actividad del proyecto: Programación utilizando estructuras repetitivas.

Estructura de contenido

- Introducción.
- Objetivos.
- Procedimientos.
- Bibliografía.



Introducción

La presente practica se realizará procedimientos y ejercicios de programación relacionado con las estructuras repetitivas con el lenguaje de programación Python usando herramientas como Google Colab o Visual Studio Code para la codificación de los programas y ejecución de pruebas.

Realizar las practicas del lenguaje de programación Python es base requerida para ser desarrollador de programas o aplicaciones con uno de los lenguajes de programación más populares. Para el desarrollo del laboratorio es necesario que el estudiante tenga disponible un computador con acceso a internet

Objetivos

Al completar esta práctica, usted podrá:

- ✓ Identificar el uso de variables contadoras y acumuladoras.
- ✓ Conocer el concepto y estructura repetitivas.
- ✓ Codificar algoritmos con condicionales y selectivas con el lenguaje de programación Python.



Procedimiento

Para el desarrollo de las practicas es necesario tener en cuenta los siguientes conceptos.

Variables contadoras

Una variable contadora en algoritmos hace precisamente esto mismo, lleva la contabilidad de repeticiones, eventos, accesos, etc... Lo importante acá es que en el contador incrementa o decremento de forma constante. Los ciclos para llevan un contador interno que mide el avance de las repeticiones.

$$con = con + 1$$

$$con = con - 1$$

Ejemplo: Suponga que está en la entrada de un evento y su jefe le pide que le diga al final cuántas personas han venido al evento

Variables acumuladoras

Un acumulador en programación es una versión ampliada de un contador. El acumulador tiene las mismas características que un contador excepto el valor de incremento que es un valor **variable**.

$$acu = acu + (valor)$$

Ejemplo: una cuenta de ahorros puede representarse en un algoritmo mediante un acumulador, pues la persona no siempre podrá ahorrar una cantidad fija en la cuenta, un día deposita 10, otro día deposita 30, otro deposita 5.



Teniendo en cuenta los conceptos de contadora y acumuladora, llenar la siguiente tabla indicando si se debe aplicar contadora o acumuladora

Situación	Tipo de variable
Cantidad de hijos	
Ahorro en un banco	
Cantidad de estudiantes en el salón	
Monto recolectado para donar	
Números de partidos ganados de futbol	
Recarga de tarjeta de transporte	

Estructuras repetitivas

Las estructuras de control repetitivas son aquellas que permiten ejecutar un conjunto de instrucciones varias veces, de acuerdo al valor que genere la expresión relacional y/o lógica. Esta estructura repetitiva se utiliza principalmente cuando no se conoce el número de veces que las acciones o el ciclo deben repetirse, aunque también puede utilizarse en otros casos. Las estructuras repetitivas, permiten repetir una acción (o grupo de acciones) varias veces.

Las estructuras repetitivas en Python son

- ✓ Estructura **for**
- ✓ Estructura **while**



Estructura **for**

for x in range(y):

x = indica el contador

y = indica el valor final (menos 1)

for x in range(p,y):

x = indica el contador

p = indica el punto de inicio

y = indica el valor final (menos 1)

Ejemplo

```
for x in range (101):
```

```
    print(x)
```

Tenemos primero la palabra clave **for** y seguidamente el nombre de la variable que almacenará en cada vuelta del **for** el valor entero que retorna la función **range**.

La función **range** retorna la primera vez el valor 0 y se almacena en **x**, luego el 1 y así sucesivamente hasta que retorna el valor que le pasamos a **range** menos uno (es decir en nuestro ejemplo al final retorna un 100)

Ejemplo

Realizar un algoritmo que permita obtener la edad de N personas entre hombres y mujeres. El valor de cada edad debe ser acumulado para los siguientes cálculos

- ✓ Cantidad de hombres y promedio de edad de hombres
- ✓ Cantidad de mujeres y promedio de edad de hombres
- ✓ Promedio general de edad.

```
1  Algoritmo sin_titulo
2      con_h←0
3      acu_h←0
4      Escribir "digite cantidad de personas"
5      leer n
6      Para i←1 Hasta n Con Paso 1 Hacer
7          Escribir "digite edad de la persona ",i
8          leer edad
9          escribir "1.- hombre o 2.- mujer"
10         leer g
11         Si g=1 Entonces
12             con_h←con_h+1
13             acu_h←acu_h+edad
14         SiNo
15             Si g=2 Entonces
16                 con_m←con_m+1
17                 acu_m←acu_m+edad
18             Fin Si
19         Fin Si
20     Fin Para
21     Si con_h=0 Entonces
22         Escribir "no se registraron hombres"
23     SiNo
24         pro_h←acu_h/con_h
25     Fin Si
26     Si con_m=0 Entonces
27         Escribir "no se registraron mujeres"
28     SiNo
29         pro_m←acu_m/con_m
30     Fin Si
31     pro_g←(acu_h+acu_m)/(con_h+con_m)
32     Escribir "cantidad de hombres : ",con_h
33     Escribir "promedio de edad de hombres : ",pro_h
34     Escribir "cantidad de mujeres : ",con_m
35     Escribir "promedio de edad de mujeres : ",pro_m
36     Escribir "promedio general: ",pro_g
37
38 FinAlgoritmo
```

```
con_h = 0
acu_h = 0
con_m = 0
acu_m = 0
pro_h = 0
pro_m = 0
pro_g = 0
n = int(input("digite cantidad de personas"))
for i in range(1,n+1):
    print("digite edad de la persona ",i)
    edad = float(input())

    g = int(input("1.- hombre o 2.- mujer"))
    if g==1:
        con_h = con_h+1
        acu_h = acu_h+edad
    else:
        if g==2:
            con_m = con_m+1
            acu_m = acu_m+edad
if con_h==0:
    print("no se registraron hombres")
else:
    pro_h = acu_h/con_h
if con_m==0:
    print("no se registraron mujeres")
else:
    pro_m = acu_m/con_m
    pro_g = (acu_h+acu_m)/(con_h+con_m)
print("cantidad de hombres : "+str(con_h))
print("promedio de edad de hombres : "+str(pro_h))
print("cantidad de mujeres : "+str(con_m))
print("promedio de edad de mujeres : "+str(pro_m))
print("promedio general: "+str(pro_g))
```



Estructura **while**

```
while x<=y:
```

En primer lugar, se verifica la condición, si la misma resulta verdadera se ejecutan las operaciones que indicamos por la rama del Verdadero.

A la rama del verdadero la graficamos en la parte inferior de la condición. Una línea al final del bloque de repetición la conecta con la parte superior de la estructura repetitiva.

En caso que la condición sea Falsa continúa por la rama del Falso y sale de la estructura repetitiva para continuar con la ejecución del algoritmo.

El bloque se repite MIENTRAS la condición sea Verdadera.

Importante: Si la condición siempre retorna verdadero estamos en presencia de un ciclo repetitivo infinito. Dicha situación es un error de programación lógico, nunca finalizará el programa.

Ejemplo

Calcular el salario de un grupo de n trabajadores dada la cantidad de horas trabajadas y la tarifa por hora para cada uno de los trabajadores. Indicar cuantos hombres y mujeres se registraron.


```
1  Algoritmo sin_titulo
2      acu←0
3      Escribir "digite cantidad de trabajadores"
4      leer n
5      i←1
6      Mientras i≤n Hacer
7          Escribir "digite cantidad de horas "
8          leer h
9          Escribir "digite valor de la hora"
10         leer v
11         sueldo←v*h
12         acu=acu+sueldo
13         escribir "1.- hombre o 2.- mujer"
14         leer g
15         Si g=1 Entonces
16             con_h←con_h+1
17         SiNo
18             Si g=2 Entonces
19                 con_m←con_m+1
20             Fin Si
21         Fin Si
22         i←i+1
23     Fin Mientras
24     Escribir "acumulado de sueldos : ",acu
25     Escribir "cantidad de hombres : ",con_h
26     Escribir "cantidad de mujeres : ",con_m
27 FinAlgoritmo
```

```
1
2  acu = 0
3  con_h=0
4  con_m=0
5
6  n = int(input("digite cantidad de trabajadores "))
7  i = 1
8  while i<=n:
9      h = float(input("digite cantidad de horas "))
10     v = float(input("digite valor de la hora "))
11     sueldo = v*h
12     acu = acu+sueldo
13
14     g = int(input("1.- hombre o 2.- mujer "))
15     if g==1:
16         con_h = con_h+1
17     else:
18         if g==2:
19             con_m = con_m+1
20     i = i+1
21 print("acumulado de sueldos : "+str(acu))
22 print("cantidad de hombres : "+str(con_h))
23 print("cantidad de mujeres : "+str(con_m))
24
```

Codificar los siguientes algoritmos en el lenguaje de programación Python

Algoritmo 1

```
Proceso contadorDeSalarios|

    c50 <- 0
    c15 <- 0

    Para x <- 1 Hasta 50 Hacer

        Leer salario

        Si (salario >= 500000 Y salario <= 1000000) Entonces

            c50 <- c50 + 1

        Sino Si ( salario > 1500000) Entonces

            c15 <- c15 + 1

        FinSi

    FinPara

    Escribir "Empleados con salario entre 500000 y 1000000: ", c50,
    " empleados con salario superior a 1500000: ", c15

FinProceso
```

Algoritmo 2

```
Algoritmo Repetitivos05

    XPROM_MAY = 0

    Para Cont <-1 Hasta 5 Con Paso 1 Hacer

        Escribir "Ingrese Nombre ", Cont , " : "
        Leer NOMBRE
        Escribir "Ingrese Promedio ", Cont , " : "
        Leer PROMEDIO

        Si PROMEDIO >= 0 Y PROMEDIO <= 20 Entonces
            Si XPROM_MAY < PROMEDIO Entonces
                XPROM_MAY = PROMEDIO
                XNOM = NOMBRE
            Fin Si
        SiNo
            CONT = CONT - 1
        Fin Si

    Fin Para

    Escribir "ALUMNO : ", XNOM
    Escribir "PROMEDIO : ", XPROM_MAY

FinAlgoritmo
```

Algoritmo 3

```
Proceso SALARIODEEMPLEADORES
Escribir 'NUMEROS DE TRABAJADORES';
Leer N;
STE<-0;
I<-0;
EGM<-100000000;
Mientras I<N Hacer
    Escribir 'SUELDO BASICO MENSUAL';
    Leer SE;
    STE<-STE+SE;
    SI EGM>SE ENTONCES
        EGM<-SE;
    FINSI
    I<-I+1;
FinMientras
Escribir 'SALARIO TOTAL DE EMPLADORES';
Escribir STE;
Escribir 'EMPLEADOR QUE GANA MENOS';
Escribir EGM;
FinProceso
```

Algoritmo 4

```
Proceso sin_titulo
    A<-0;
    Mientras A>10 Hacer
        A<-A+1;
        B<-0;
        Mientras B>10 Hacer
            B<-B+1;
            Escribir A, 'X', B, '=', 'A*B';
        FinMientras
    FinMientras
FinProceso
```

Algoritmo 5

Algoritmo Repetitivo11

Escribir "Ingrese Números de 3 Cifras"

Para Cont <-1 Hasta 30 Con Paso 1 Hacer

Escribir "Ingrese Número " Cont " : "
Leer Num

Si Num <= 999 Entonces

C1 = (Num - (Num MOD 100)) / 100
R1 = Num MOD 100
C2 = (R1 - (R1 MOD 10)) / 10
R2 = R1 MOD 10

Escribir R2 " " C2 " " C1

SiNo

Cont = Cont + 1

Fin Si

Fin Para

FinAlgoritmo



Referencias Bibliográficas

Universia. (2017). ¿Qué es y para qué sirve Python? Retrieved May 21, 2019, from <http://noticias.universia.es/ciencia-tecnologia/noticia/2017/07/19/1154393/sirve-python.html>

Ojeda, L. R. (2017). *Escuela Superior Politécnica del Litoral Python Programación Libro digital Versión 3.0-2017*. Retrieved from http://blog.espol.edu.ec/icm00794/files/2017/05/PYTHON_PROGRAMACION_V3_0.pdf