



Unidad Azcapotzalco

División de Ciencias y Artes para el Diseño

Posgrado en Diseño y Visualización de la Información

Estructuras repetitivas

L. C. C. Eric Omar Torres Velasco

Ensayo de investigación para la UEA Introducción a la programación

Profesora:

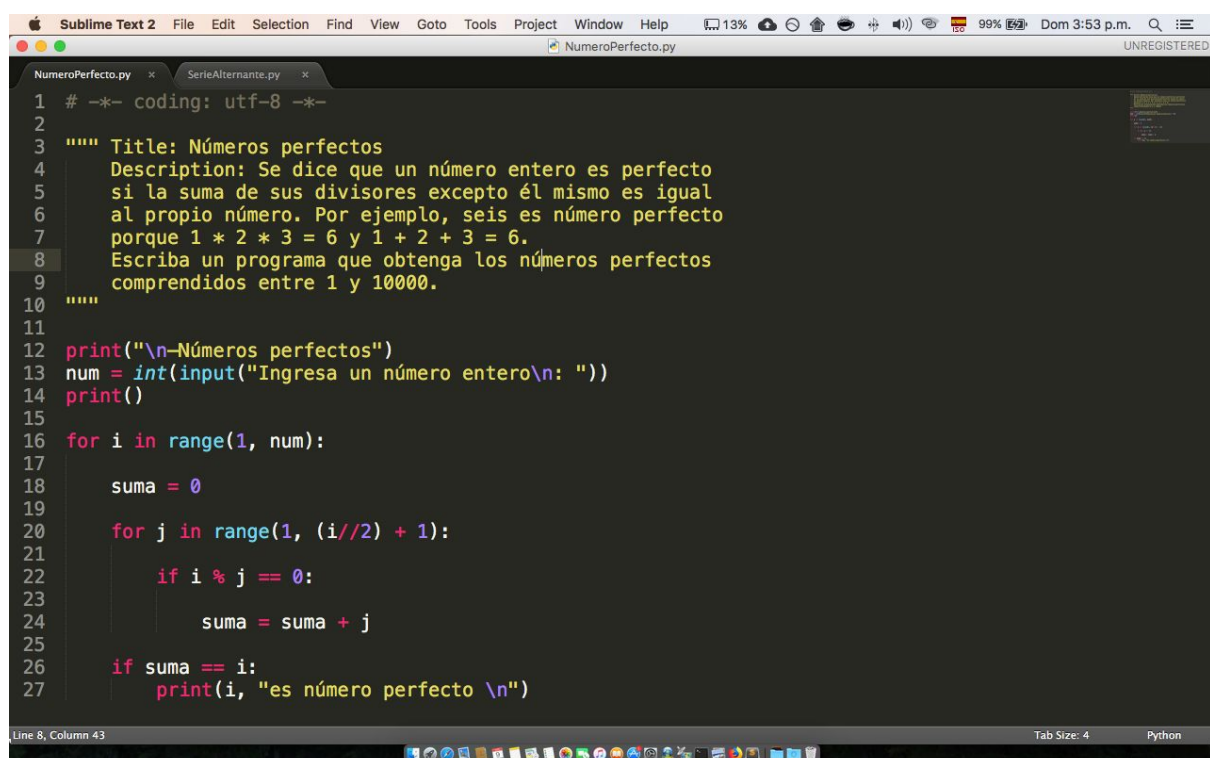
Dra. Lizbeth Gallardo López

Ciudad de México, México. A 9 junio de 2019

En computación, una estructura repetitiva o iterativa es una instrucción que repite un determinado número de veces, uno o varios bloques de sentencias. La cantidad de veces que opera mientras se ejecuta el programa, depende de una condición o expresión lógica establecida. En programación, los ciclos más comunes son: *for* (para), *while* (mientras) y *do-while* (repetir-hasta). “A este tipo de sentencias también se le conocen como sentencias de bucle (loop)” (González, 2007: 42).

La estructura *for* es el ciclo que repite un grupo de sentencias, mientras una condición se cumpla, a partir de la inicialización de un valor índice, que incrementa o decrementa durante su ejecución, hasta no exceder de un valor final o, en efecto, el ciclo terminaría. Por lo general, este bucle “se utiliza en soluciones donde se conoce el número de veces que deberá repetirse el ciclo” (Gallardo & Lara, 2012: 41).

A continuación, se presenta un ejemplo de la estructura repetitiva *for*, mediante la sintaxis de python, para un programa que muestra los números perfectos comprendidos entre 1 y un valor numérico entero ingresado por el usuario:

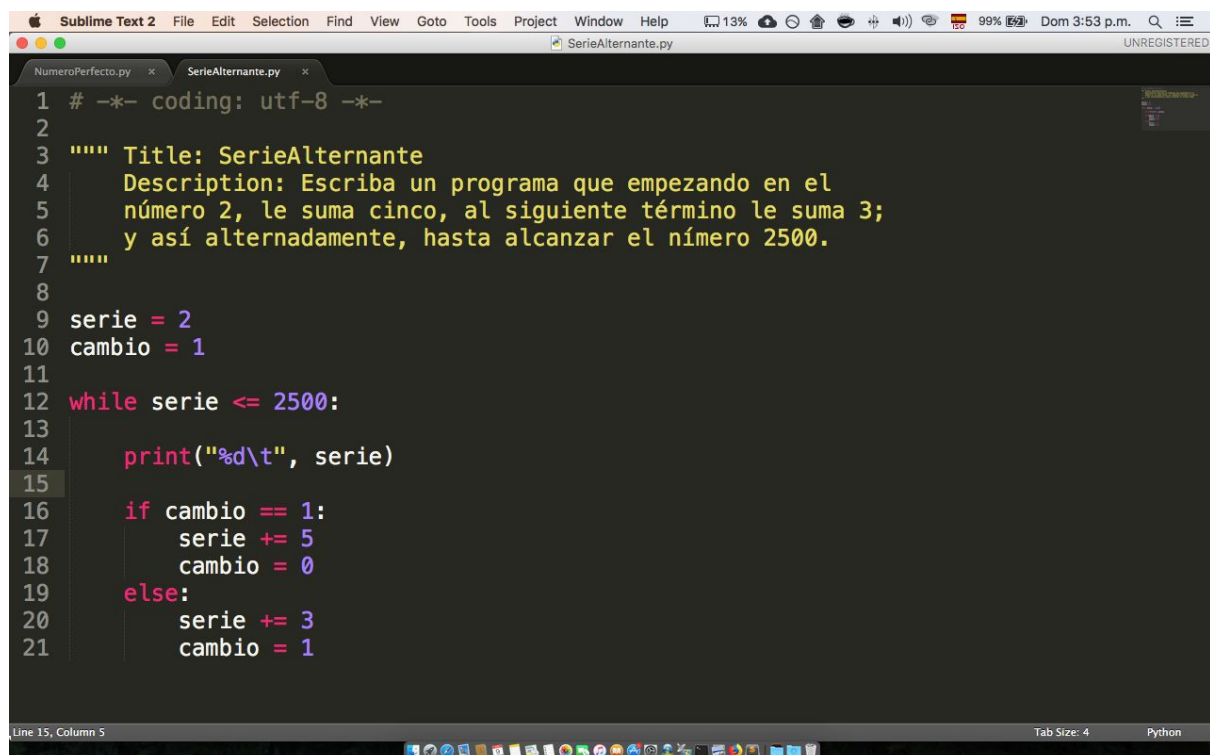


```
1  # -*- coding: utf-8 -*-
2
3  """ Title: Números perfectos
4      Description: Se dice que un número entero es perfecto
5      si la suma de sus divisores excepto él mismo es igual
6      al propio número. Por ejemplo, seis es número perfecto
7      porque 1 * 2 * 3 = 6 y 1 + 2 + 3 = 6.
8      Escriba un programa que obtenga los números perfectos
9      comprendidos entre 1 y 10000.
10 """
11
12 print("\nNúmeros perfectos")
13 num = int(input("Ingresa un número entero\n: "))
14 print()
15
16 for i in range(1, num):
17     suma = 0
18     for j in range(1, (i//2) + 1):
19         if i % j == 0:
20             suma = suma + j
21
22     if suma == i:
23         print(i, "es número perfecto \n")
```

Ejemplo 1. Sintaxis de estructura *for* en python.

Una estructura *while* es el bucle que repite las instrucciones mientras la expresión lógica que se evalúa sea verdadera, es decir, si la condición es verdadera las sentencias se ejecutan N número de veces, pero si es falsa no se ejecutan y se continúa con el flujo del programa. Por lo general, este loop “se utiliza para repetir un bloque de instrucciones durante un número indeterminado de veces” (Martínez, 2013: 14).

A continuación, se muestra un ejemplo de la estructura iterativa *while*, con la sintaxis de python, para un programa que ejecuta una serie comenzando por el número 2 y que incrementa primero 5 y luego 3, después, otra vez 5 y luego, otra vez, 3 y así, hasta alcanzar el número 2500:



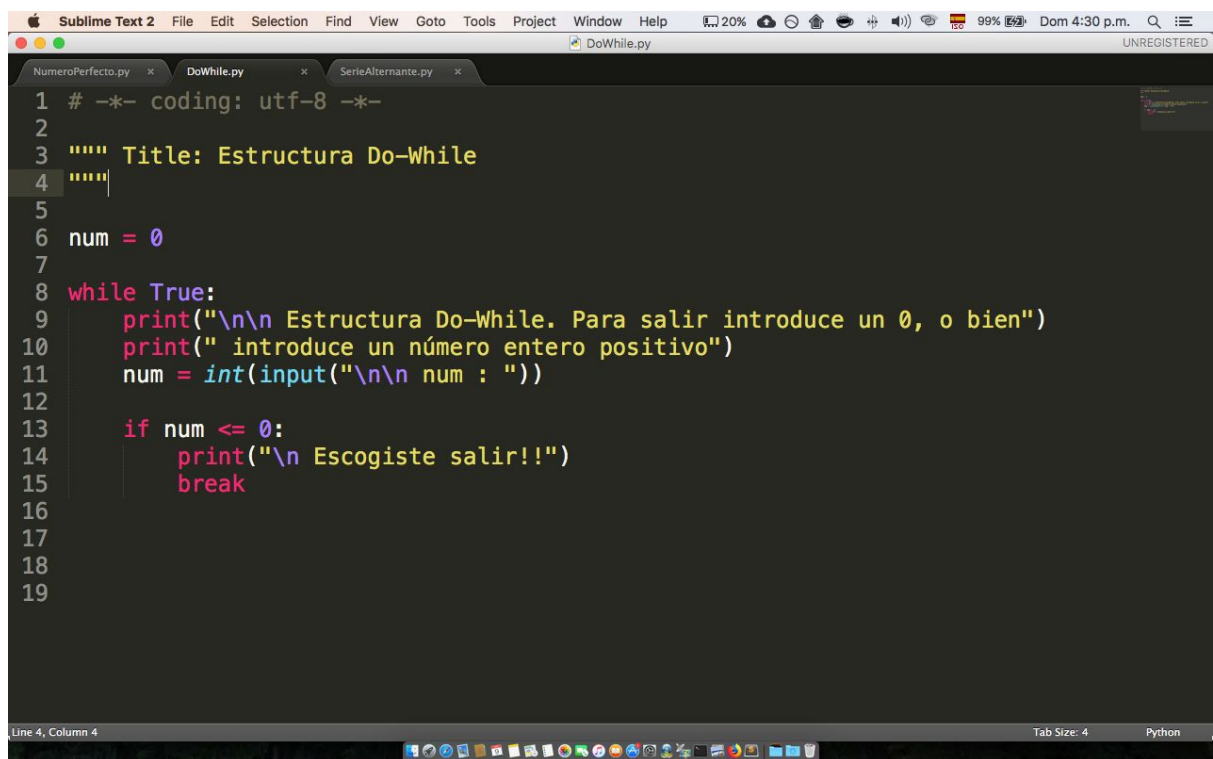
```
1 # -*- coding: utf-8 -*-
2
3 """ Title: SerieAlternante
4     Description: Escriba un programa que empezando en el
5     número 2, le suma cinco, al siguiente término le suma 3;
6     y así alternadamente, hasta alcanzar el número 2500.
7 """
8
9 serie = 2
10 cambio = 1
11
12 while serie <= 2500:
13
14     print("%d\t", serie)
15
16     if cambio == 1:
17         serie += 5
18         cambio = 0
19     else:
20         serie += 3
21         cambio = 1
```

Ejemplo 2. Sintaxis de estructura *while* en python.

La estructura *do-while* “es una variación del ciclo *while* y en esta estructura se ejecutan las sentencias y luego se evalúa la condición o expresión lógica, si es falsa se ejecutan las sentencias N veces, si es verdadera se sale del ciclo” (González, 2007: 45). Esto es, el loop que por lo menos ejecuta una repetición de bloque de

instrucciones y hasta el final comprueba si la condición se cumple o no. “La diferencia con el ciclo *while* consiste en la certeza de que el bloque se ejecutará al menos una vez” (Martínez, 2013: 14).

A continuación, se expone un ejemplo de la estructura de bucle *do-while*, usando la sintaxis de python, para un programa que ejecuta un bloque de código hasta que el usuario introduzca un número entero menor o igual a 0:



```
1 # -*- coding: utf-8 -*-
2
3 """ Title: Estructura Do-While
4 """
5
6 num = 0
7
8 while True:
9     print("\n\n Estructura Do-While. Para salir introduce un 0, o bien")
10    print(" introduce un número entero positivo")
11    num = int(input("\n\n num : "))
12
13    if num <= 0:
14        print("\n Escogiste salir!!")
15        break
16
17
18
19
```

Ejemplo 3. Sintaxis de estructura *do-while* en python.

Referencias y bibliografía:

Gallardo, Lizbeth & Lara, Pedro. (2012). Notas de programación estructurada. Universidad Autónoma Metropolitana.

González, Oscar. (2007). Introducción a la programación. México: Universidad Autónoma de México.

Martínez, Guillermo. (2013). Lenguaje de programación Java. México: Universidad Nacional Autónoma de México.