

# Actividad 2

Carolina Valenzuela Córdova

28 de Enero de 2016

En esta actividad, se nos proporcionaron modelos de códigos en Python para que realizaran ciertas operaciones matemáticas y nos dieran resultados diferentes. Utilizamos una plataforma en línea para poder programar con Python y correr los programas que modificamos. En la actividad se indicaron 5 programas diferentes y a continuación se brinda su descripción y las capturas de pantalla pertinentes.

```
h = float(input("Proporciona la
altura
de la torre: "))
t = float(input("Ingresa el
tiempo:"))
s = 0.5*9.81*t**2
print("La altura de la pelota es",
h-s,
"metros")
```

Se tuvo que reemplazar la ecuación utilizada en el código original por una que calculara el tiempo de caída de la pelota:

$$t = \sqrt{\frac{2h}{g}}$$

## 1. Problema 1: Caída libre

Se nos proporcionó un código que calcula la distancia que recorre una pelota cuando se deja caer desde el techo de una torre, es decir, calcula la altura de la misma. La instrucción era modificarlo para que solicitara al usuario la altura de la torre en metros, y entonces que imprimiera el tiempo que tarda la pelota en llegar al suelo, ignorando la fricción del aire. El código original era el siguiente:

, siendo  $g = 9.81m/s^2$ . También cambiamos el hecho de que el programa solicitara el tiempo, siendo ahora solo requerida la altura de la torre. El programa ya modificado resultó así:

```
h = float(input("Proporciona la altura
de la torre: "))
t = sqrt((2*h)/9.81)
print ("El tiempo que tarda la pelota
en caer son",t,"segundos")
```

y podemos observar los resultados:

Proporciona la altura de la torre: 20  
(‘El tiempo que tarda la pelota en caer  
son’, 2.0192751093846089, ‘segundos’)

## 2. Problema 2: Satélite

Se nos solicitó elaborar un programa que calculara las altitudes para que un satélite orbitara la Tierra una vez al día, cada 90 y 45 minutos, a partir de pedirle al usuario que ingresara el valor de  $T$ , de tal manera que le regresara una altura  $h$  en metros. Se nos proporcionó la siguiente ecuación:

$$(R + h)^3 = \frac{GMT^2}{4\pi^2}$$

de la cual, despejando  $h$  obtuvimos

$$h = \sqrt[3]{\frac{GMT^2}{4\pi^2}} - R$$

Esta fue la ecuación que insertamos en el código. Además, tuvimos que realizar algunas conversiones para que todas las unidades coincidieran, teniendo como valores constantes  $G = 6.67 \times 10^{-11} \text{m}^3 \text{kg}^{-1} \text{s}^{-2}$ , (constante de Gravitación Universal de Newton),  $M = 5.97 \times 10^{24} \text{kg}$  (masa de la Tierra) y  $R = 6371000 \text{m}$  (radio de la Tierra), además, el periodo que ingresa el usuario se solicita en segundos.

```
from math import pi
T= float(input("Ingrese un valor para
el periodo en segundos"))
G=6.67e-11
M=5.97e24
R=6371000
h=(((G*M*T*T)/(4*pi*pi))**(1./3.))-R
print ("El valor de la altitud del
satelite sobre la superficie de la
Tierra es",h, "metros")
```

Aquí se muestran los resultados obtenidos para 45 y 90 minutos, es decir 2700 y 5400 segundos, respectivamente.

```
Ingrese un valor para el periodo en
segundos2700
('El valor de la altitud del satelite
sobre la superficie de la Tierra es',
-2181559.8978108233, 'metros')
```

```
Ingrese un valor para el periodo en
segundos5400
('El valor de la altitud del satelite
sobre la superficie de la Tierra es',
279321.6253728606, 'metros')
```

## 3. Problema 3: Coordenadas polares

Se proporcionó un código para transformar coordenadas cartesianas a partir de coordenadas polares:

```

from math import sin,cos,pi
r = float(input("Introduce r: "))
d = float(input("Ingresa theta en
grados: "))
theta = d*pi/180
x = r*cos(theta)
y = r*sin(theta)
print("x =",x," y =",y)

```

Se solicitó hacer un código similar para calcular coordenadas esféricas a partir de cartesianas. Para ello fue necesario encontrar una relación entre estos dos tipos de coordenadas y esta fue

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

$$\varphi = \arctan \frac{y}{x}$$

Sabiendo esto, pudimos escribir un código que realizara estas operaciones solicitándole al usuario que ingresara valores para  $x$ ,  $y$  y  $z$ . Nuestro código resultó de la siguiente manera:

```

from math import sin,cos,pi,
acos,atan
x = float(input("Introduce
un valor
para x: "))
y = float(input("Introduce
un valor
para y:"))
z = float(input("Introduce
un valor
para z:"))
r=sqrt((x**2)+(y**2)+(z**2))
theta=acos((z/r))
phi=atan((y/x))

print("r=",r," theta =",theta,"phi="
,phi)

```

Resultados obtenidos al correr el programa:

```

Introduce un valor para x: 1
Introduce un valor para y:2
Introduce un valor para z:1
('r=', 2.4494897427831779, ' theta =',
1.1502619915109313, 'phi=',
1.1071487177940904)

```

#### 4. Problema 4: Números pares e impares

En este ejercicio solo se solicitó correr un programa que detecta números pares e impares y comprender su dinámica, pues utiliza el comando *while*. El programa es el siguiente:

```

print("Ingrese dos números,
un par y un
impar.")
m = int(input("Ingrese
el primero: "))
n = int(input("Ingrese
el segundo: "))
while (m+n)%2==0:
    print("Uno debe ser
    par y otro
    impar.")
    m = int(input("Ingrese
    el primer
    número: "))
    n = int(input("Ingrese
    el segundo
    número: "))
print("Los numeros que
ligio son ",m,"y",n)

```

Pudimos observar que el comando se utiliza para realizar ciclos, los cuales sirven para resolver ecuaciones en repetidas ocasiones con diferentes valores que se establecen en el mismo programa, a esto comúnmente se le llama *iterar*. La comprensión del comando nos permitió realizar con mayor facilidad el siguiente programa.

## 5. Problema 5: Secuencia de números de Catalán

Se nos solicitó escribir un código que calculara la secuencia de números de Catalán menores o igual que 1,000,000. Fue más sencillo realizar este código porque ya habíamos comprendido el uso del comando *while* y además nos dieron un ejemplo similar que concierne a

la sucesión de números Fibonacci. El código que usamos como ejemplo fue:

```

f1,f2 = 1,1
while f2<1000:
    print(f2)
    f1,f2 = f2,f1+f2

```

Se nos proporcionó de igual manera una ecuación que calculara la sucesión de números de Catalán, la cual es

$$C_{n+1} = \frac{2(2n+1)}{n+2}C_n$$

Así, con conocimiento de estos datos pudimos realizar el programa, el cual resultó ser:

```

n,c=0.,1.
while c<=1000000:
    print(c)
    n,c = (n+1),(2*(2*n+1)
    /(n+2))*c

```

Resultados obtenidos al correr el programa:

```

1.0
1.0
2.0
5.0
14.0
42.0
132.0
429.0
1430.0
4862.0
16796.0
58786.0
208012.0
742900.0

```

## **6. Conclusión**

Me pareció muy importante la realización de esta práctica, pues pudimos comenzar a tener contacto con diferentes tipos de programas que se pueden realizar con Python. Además pudimos comprender el funcionamiento de algunos comandos nuevos y la estructura de los códigos en este tipo de compilador. Podemos notar de igual manera que nuestra experiencia con programación nos ha ayudado a comprender con mayor facilidad los procedimientos para crear códigos, así como para hacer reportes con mayor calidad, gracias a la práctica anterior.