

Taller de Ejercicios Python

Erick Sebastian Mora Lara^[L00082852]

Universidad de las Fuerzas Armadas
esmora2@espe.edu.ec

Abstract.

Dentro del presente trabajo se plantean dar énfasis y resolución a varios problemas algorítmicos varios expuestos dentro de la materia de Modelos Discretos para Ingeniería, fundamentado en el Análisis, Modelado y Desarrollo de los mismos. Para lo cual se usaran modelos para la resolución de problemas como el modelo de los 6 pasos, describiendo, planteando, desarrollando y ejecutando los programas, de tal manera que su uso sea legible para el usuario lector y para la computadora mediante el empleo del lenguaje de programación Python

1 Introducción

Dentro de la Informática y el desarrollo de Software y programas de computadora, existen un gran número de Lenguajes o medios para alcanzar el objetivo de desarrollar algoritmos con diferentes propósitos. Tenemos al alcance varios lenguajes de programación unos más potentes que otros, así mismo la sintaxis de algunos será más sencilla y con una comprensión más fácil que otros.

Python es un lenguaje de programación de alto nivel y de propósito general. Fue creado a finales de los años 80 por Guido van Rossum y ha conseguido ganar una gran popularidad en los últimos años debido a su sintaxis sencilla y su amplia gama de aplicaciones.

Según [2] uno de los usos más comunes de Python es el empleo del lenguaje para resolver problemas matemáticos y científicos. Python incluye una gran cantidad de bibliotecas y módulos que facilitan el cálculo y el análisis de datos. Por ejemplo, la biblioteca NumPy proporciona funciones y herramientas para trabajar con matrices y hacer cálculos matriciales, mientras que la biblioteca SciPy ofrece una amplia gama de funciones matemáticas y estadísticas.

Además de sus aplicaciones matemáticas, Python también se utiliza ampliamente en el desarrollo de aplicaciones web, el análisis de datos y la automatización de tareas. Debido a su gran comunidad de desarrolladores y su amplia base de código abierto, Python es una opción popular para muchos tipos de proyectos.

En resumen, Python es un lenguaje de programación versátil y poderoso que se utiliza ampliamente para resolver problemas matemáticos y científicos, así

como para muchas otras aplicaciones. Con su sintaxis sencilla y su gran cantidad de bibliotecas y módulos disponibles, Python es una opción atractiva para muchos tipos de proyectos. Y para el trabajo planteado el lenguaje Python sera nuestro medio para desarrollar y dar solucion a nuestros problemas.

2 Desarrollo

Aplicando los 6 pasos de resolución de problemas, se van desarrollar a continuación cada uno de los ejercicios planteados.

2.1 Ejercicio 1: Perímetro y Área del Rectángulo

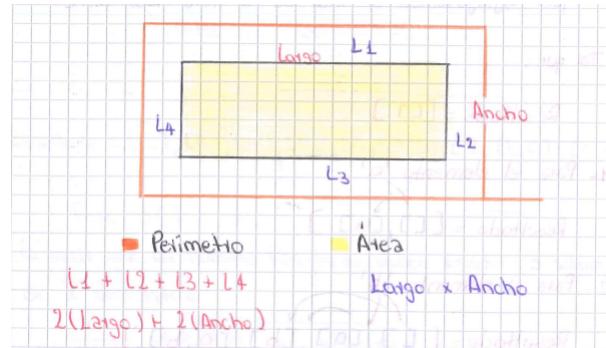
Dadas dos longitudes ingresadas por el usuario que corresponden a los lados de un rectángulo, calcular el perímetro y el área del mismo.

Comprensión del problema: El rectángulo es una figura geométrica de la clasificación de los paralelogramos que al igual que otras formas geométricas, cuentan con sus dimensiones. Particularmente, si nos referimos a los rectángulos estos tienen las suyas bien definidas y las conocemos como largo y ancho, dado que conocemos que el rectángulo es una composición de 4 lados y 2 de ellos son iguales respectivamente a sus lados paralelos.

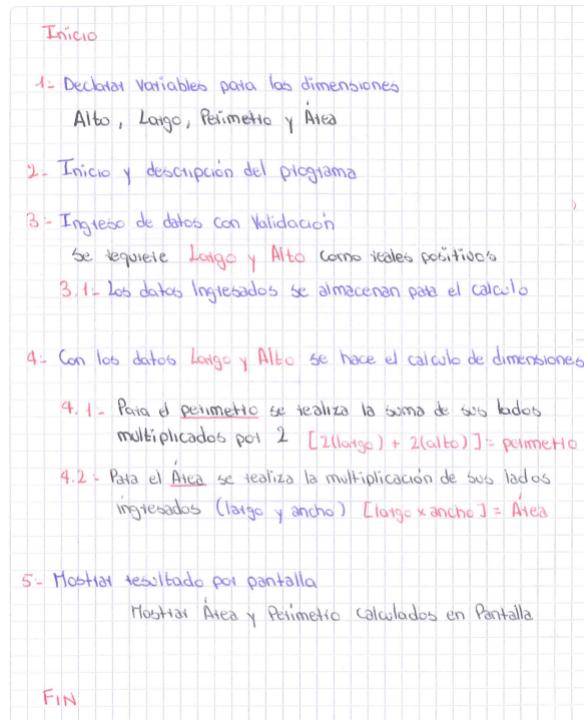
Sin embargo, los lados no son los únicos datos que se pueden obtener dado un rectángulo definido, a partir de sus lados conocidos se pueden calcular dimensiones como su área y su perímetro. El perímetro no es más que la suma de las dimensiones de los 4 lados de la figura o lo que es similar el doble de su largo más el doble de su ancho.

En cuanto al área nos referimos, no es más que la dimensión de la zona interna del rectángulo encerrada entre sus 4 lados, su cálculo no supone complejidad alguna y únicamente basta con calcular el lado más largo multiplicado por el lado más ancho.

Formular un modelo: Para poder implementar un modelo algorítmico que calcule los datos de un rectángulo haciendo uso del lenguaje Python, es necesario tener claro que es lo que tenemos y que es lo que buscamos. El problema dicta que debemos conocer dos lados de un rectángulo, dado que la condición de un rectángulo es que únicamente tiene un lado ancho y uno largo y dichos lados se replican paralelamente para encerrar a la figura geométrica podemos asegurar que, el solo conocer estas dos dimensiones será información suficiente para realizar los cálculos pertinentes para el problema.

**Fig. 1.** Analisis de los Datos de un Rectangulo

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

**Fig. 2.** Pseudocodigo del Proceso para obtener Area y Perimetro

Escribe el programa:

En el Listing 1.1. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular los valores de área y perímetro de un rectángulo habiendo ingresado sus lados (ancho y largo).

```

1      """
2      """
3      Descripcion :
4      Ejercicio 1.
5      Perimetro y Area del Rectangulo
6      Dadas dos longitudes ingresadas por el usuario que
7      corresponden a los lados de un rectangulo ,
8      calcular el perimetro y el area del mismo.
9      Autor:
10     Erick Sebastian Mora Lara
11     """
12
13     def ingresarLados():
14         """
15             Es un procedimiento que solicita el ingreso de los
16             lados por teclado ,
17             se combina con la funcion validarNumeros() , para
18             validar únicamente
19             el ingreso de datos numéricos
20
21             _____
22             No Recibe parámetros de entrada
23
24             Retorna :
25             _____
26             Retorna una lista que contiene los dos lados
27             ingresados (lado1 , lado2)
28             """
29             ladosLista = [] #Definir lista vacia para almacenar el
30             largo y ancho del rectangulo
31             print("Vamos_a_Calcular_el_Perimetro_y_Area_de_un_
32             Rectangulo:")
33             print("Necesitas_conocer_la_dimension_del_Largo_y_
34             Ancho")
35             #Mensaje de Inicio
36             #Ciclo For que iterá dos veces para solicitar el
37             #Ingreso de dos datos por teclado
38             for i in range(2):
39                 print("Ingrese_el_lado_ " , i+1, " : ")
40                 lados= validarNumeros() #El ingreso se fusiona con
41                 la validacion de numeros reales mayores a 0
42                 ladosLista.append(lados) #Se almacenan ambos lados
43                 en una lista y se retorna dicha lista para el
44                 calculo en las funciones
45             return ladosLista
46
47

```

```
35  def validarNumeros():
36      """
37          Es un procedimiento generico alternativo para realizar
38          validacion de numeros se lee un tipo de
39          dato ingresado y se intenta su conversion a flotante (
40              float), si es numero se transforman y pasa a la
41          validacion para verificar que el numero sea mayor que
42              0.
43          Si la conversion falla, se emplea el uso de
44              Excepciones para imprimir un mensaje de Error
45
46      No Recibe parametros de entrada
47
48      Retorna:
49
50          Retorna un valor entero (numero) que es el dato
51          ingresado transformado a (int)
52      """
53
54  while True:
55      num = input()
56      try:
57          num = float(num)
58          if num >0:
59              return num
60          else:
61              print("Solo numeros mayores a 0")
62      except ValueError:
63          print("Solo se aceptan Reales")
64
65  def perimetroRectangulo(lado1, lado2):
66      """
67          Es un procedimiento que recibe como datos dos lados de
68          un rectangulo
69          y con ellos calcula su perimetro haciendo la suma de
70          sus dos lados multiplicados por 2
71
72          Recibe como parametros dos lados de un rectangulo,
73          validados para que sean solo de tipo numerico
74
75      Retorna:
76
77          Retorna el valor del perimetro obtenido mediante
78          la operacion de suma
79      """
80
81  perimetroRec = lado1*2+lado2*2
82  print("El perimetro del Rectangulo Ingresado es: ", perimetroRec)
83
84  def areaRectangulo(lado1, lado2):
85      """
```

```

75      Es un procedimiento que recibe como datos dos lados de
76          un rectangulo
77      y con ellos calcula su area haciendo la multiplicacion
78          de ambos
79
80      Recibe como parametros dos lados de un rectangulo ,
81          validados para que sean solo de tipo numerico
82
83      Retorna :
84
85      Retorna el valor del area obtenido mediante la
86          operacion
87      """
88
89      Funcion Main, aqui se declararan las variables
90          principales y se llamara a las funciones
91          necesarias para
92          ejecutar el programa
93
94      """
95
96      ancho , largo = ingresarLados() #Se declaran dos
97          variables correspondientes a los lados
98      #Con ambas variables ingresadas y validadas se llama a
99          las funciones de Perimetro y Area
100
101     perimetroRectangulo(ancho , largo)
102
103     areaRectangulo(ancho , largo)

```

Listing 1.1. Desarollo del ejercicio 1

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema. Véase 4 , 5 , 6

Largo	Ancho	Perimetro	Area	
5	3	0	0	ENTRADA
5	3	$2(5)+2(3)$	$5*3$	1 ITERACION
5	3	16	15	RESULTADO

Table 1. Caso de Prueba 1

Largo	Ancho	Perimetro	Area	
18	4	0	0	ENTRADA
18	4	$2(18)+2(4)$	$18*4$	1 ITERACION
18	4	44	72	RESULTADO

Table 2. Caso de Prueba 2

Largo	Ancho	Perimetro	Area	
10	6	0	0	ENTRADA
10	6	$2(10)+2(6)$	$10*6$	1 ITERACION
10	6	32	60	RESULTADO

Table 3. Caso de Prueba 3

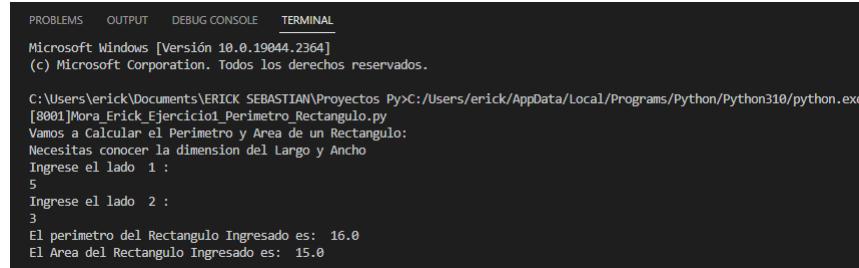
Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 4

Largo	Ancho	Perimetro	Area	
5	3	-	-	ENTRADA
5	3	$2(5)+2(3)$	$5*3$	1 ITERACION
5	3	16	15	RESULTADO

Table 4. Caso de Prueba 1

Se evaluan los datos y se obtienen los siguientes resultados. Véase 3



```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
[8001]Mora_Erick_Ejercicio1_Perimetro_Rectangulo.py
Vamos a Calcular el Perímetro y Area de un Rectángulo:
Necesitas conocer la dimension del Largo y Ancho
Ingrese el lado 1 :
5
Ingrese el lado 2 :
3
El perimetro del Rectangulo Ingresado es: 16.0
El Area del Rectangulo Ingresado es: 15.0

```

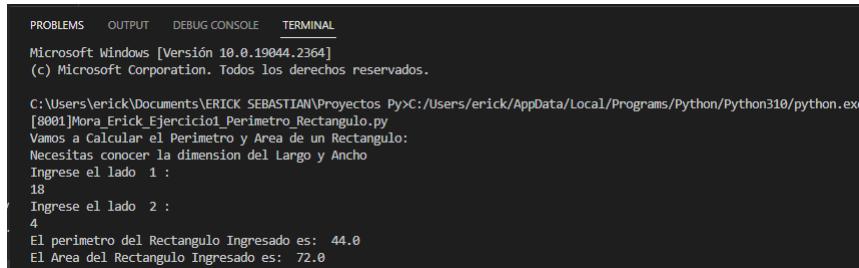
Fig. 3. Ejecucion 1: Ejercicio 1

Para el Caso de Prueba 2. 5

Largo	Ancho	Perimetro	Area	
18	4	-	-	ENTRADA
18	4	$2(18)+2(4)$	$18*4$	1 ITERACION
18	4	44	72	RESULTADO

Table 5. Caso de Prueba 2

Se evaluan los datos y se obtienen los siguientes resultados. Véase 4



```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
[8001]Mora_Erick_Ejercicio1_Perimetro_Rectangulo.py
Vamos a Calcular el Perímetro y Area de un Rectángulo:
Necesitas conocer la dimension del Largo y Ancho
Ingrese el lado 1 :
18
Ingrese el lado 2 :
4
El perimetro del Rectangulo Ingresado es: 44.0
El Area del Rectangulo Ingresado es: 72.0

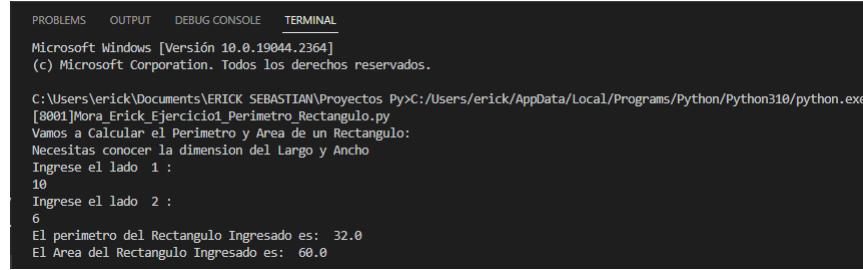
```

Fig. 4. Ejecucion 2: Ejercicio 1

Para el Caso de Prueba 3. 5

Se evaluan los datos y se obtienen los siguientes resultados. Véase 5

Largo	Ancho	Perimetro	Area	
10	6	-	-	ENTRADA
10	6	$2(10)+2(6)$	$10*6$	1 ITERACION
10	6	32	60	RESULTADO

Table 6. Caso de Prueba 3


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos PyxC:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
[8001 Mora_Erick_Ejercicio1_Perimetro_Rectangulo.py
Vamos a Calcular el Perimetro y Area de un Rectangulo:
Necesitas conocer la dimension del Largo y Ancho
Ingrese el lado 1 :
10
Ingrese el lado 2 :
6
El perimetro del Rectangulo Ingresado es: 32.0
El Area del Rectangulo Ingresado es: 60.0

```

Fig. 5. Ejecucion 3: Ejercicio 1

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, los cuales deben proporcionar una similitud total con los obtenidos en las pruebas manuales realizadas en el Análisis. Si la evaluación es correcta el ejercicio cumple con su objetivo y por lo tanto se lo da por realizado.

2.2 Ejercicio 2: Calculo de una expresión matemática

Dada la expresión matemática $y = x^2/2$, solicitar al usuario el ingreso de un valor para X y calcular el equivalente al valor en Y de acuerdo al cálculo de la expresión algebraica. .

Comprensión del problema: A continuación se nos ha planteado un ejercicio matemático cuyo problema supone ser ni más ni menos que una ecuación algebraica con dos variables tal que tenemos a una de ellas (y) igualada al cuadrado de la otra (x^2) y dividida para una constante (2). El objetivo del problema entonces es calcular el valor de Y en la expresión, para lo cual supondríamos un valor cualquiera en X para calcular la expresión, dicho valor debería ser proporcionado por el usuario que desee hallar los valores de Y en la ecuación a partir de la entrada de un valor perteneciente al rango de los Números Reales.

La resolución del problema no supone complejidad alguna y por su parte el cálculo no presenta ningún tipo de restricción de valores para el cual se deba hacer una especificación en los valores del rango. Cualquier valor del rango de los Reales que sea ingresado en la variable (X) mostrara su imagen en la variable

(Y), y esto se reflejara en el resultado.

Formular un modelo: Para formar un modelo algorítmico que desarrolle la operación algebraica en cuestión partimos del análisis realizado anteriormente y nos fijamos en cada uno de los datos que tenemos en la ecuación.

The image shows handwritten mathematical analysis on a grid paper background. At the top, it says "Calculo de la expresión" followed by the equation $y = \frac{x^2}{2}$. Below this, it says "Valor buscado" followed by the same equation $y = \frac{x^2}{2}$, with a red annotation "Variable necesaria para el cálculo" pointing to the variable x^2 . At the bottom right, there is a red arrow pointing to the number 2 with the label "Constante".

Fig. 6. Análisis de la Expresión matemática

A partir de estos datos podemos determinar el funcionamiento y el camino que va a tomar nuestro programa para alcanzar el resultado deseado. Como podemos observar en la figura, nuestra expresión matemática cuenta con tres términos: el primero y el que más nos interesa (dado que es el que manipularemos) es el término X, dicho dato se encuentra elevado a una potencia 2 (Al cuadrado). El segundo término es una constante C=2 que al igual que la potencia mencionada anteriormente no afectara a la primera variable X, pero si será indispensable para determinar el valor de nuestra variable solución (Y). El último término será la variable solución (Y), la cual requiere de la constante (segundo término) y de que el primer término se encuentre definido para conocer su valor.

En resumen, el propósito de nuestro programa es encontrar un valor para (Y) en la expresión, dado un valor en (X) cualquiera y este último será definido e ingresado por el usuario en el programa.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

Inicio

1- Declarar variables para los terminos
 x, y

2- Inicio y descripción del programa

3- Ingreso de Datos con validación
 x debe ser un Real

3.1- El ingreso se almacena en x

4- Se plantea la formula $y = x^2/2$

4.1- La formula se almacena en y

5- Se imprime el valor de y por pantalla

Fin

Fig. 7. Pseudocódigo del Proceso para obtener el resultado de la expresión matemática

Escribe el programa:

En el Listing 1.2. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular la expresión $y = x^2/2$ a partir de un valor x conocido e ingresado por teclado.

```

1
2      """
3  Descripcion:
4  Ejercicio 2.
5  Calculo de una expresion matematica
6  Realizar el calculo de la siguiente expresion  $y=x^2/2$ 
7  Autor:
8  Erick Sebastian Mora Lara
9  """
10 def ingresoDatos():
11     """
12     Es un procedimiento que solicita el ingreso de datos
13     por teclado,
14     se combina con la funcion validarNumeros(), para
15     validar únicamente
16     el ingreso de datos numéricos positivos y negativos
17
18     No Recibe parametros de entrada
19     _____

```

```
20      Retorna una variable con un valor numerico
21      ingresado por teclado
22      """
23      print(" ***VAMOS_A_CALCULAR_EL_VALOR_DE_Y_EN_LA_
24          SIGUIENTE_EXPRESION: _y=x^2/2 _***")
25      print(" Para_lo_cual_necesitas_Ingresar_un_valor_en_X: _"
26          )
27      xValor=validarNumeros()
28      return xValor
29  def validarNumeros():
30      """
31          Es un procedimiento generico alternativo para realizar
32          validacion de numeros enteros
33          los datos ingresados se leen como string (str) y si
34          son numeros se verifican
35          y se transforman en tipo de datos entero (int), en
36          este variante de funcion
37          se maneja el uso de una excepcion para el ingreso de
38          datos que no son numericos:
39
39      No Recibe parametros de entrada
40
41      Retorna:
42
43      Retorna un valor entero (numero) que es el dato
44          ingresado transformado a (int)
45      """
46  while True:
47      num = input()
48      try:
49          num = float(num)
50          return num
51      except ValueError:
52          print(" Solo_se_aceptan_Reales")
53
54  def calculoMat():
55      """
56          Es un procedimiento que realiza el calculo de la
57          expresion matematica
58          calcula el valor de Y, a partir de un valor X
59          ingresado
56
57      No Recibe parametros de entrada
58
59      Retorna:
60
61      No retorna Ningun Valor
62      """
63      xValor = ingresoDatos()
64      yResultado = ((xValor**2)/2)
```

```

60     print("x=_",xValor)
61     print("y=_",yResultado)
62
63
64 if __name__ == '__main__':
65     calculoMat()

```

Listing 1.2. Desarollo del ejercicio 2

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema. Véase 10 , 11 , 12

X	Y	$y = x^2/2$	
6	-	$y = x^2/2$	ENTRADA
6	18	$y = (6)^2/2$	1 ITERACION
6	18	$y = 18$	RESULTADO

Table 7. Caso de Prueba 1

X	Y	$y = x^2/2$	
15	-	$y = x^2/2$	ENTRADA
15	112.5	$y = (15)^2/2$	1 ITERACION
6	112.5	$y = 112.5$	RESULTADO

Table 8. Caso de Prueba 2

X	Y	$y = x^2/2$	
4.2	-	$y = x^2/2$	ENTRADA
4.2	8.82	$y = (4.2)^2/2$	1 ITERACION
4.2	8.82	$y = 8.82$	RESULTADO

Table 9. Caso de Prueba 3

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos con las variables utilizadas dentro de nuestros 3 ensayos de escritorio manuales, el resultado obtenido en la ejecución debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 10

X	Y	$y = x^2/2$	
6	-	$y = x^2/2$	ENTRADA
6	18	$y = (6)^2/2$	1 ITERACION
6	18	$y = 18$	RESULTADO

Table 10. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 8

```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py\Ejercicio2_CalcularExpresion.py"
***VAMOS A CALCULAR EL VALOR DE Y EN LA SIGUIENTE EXPRESION: y=x^2/2 ***
Para lo cual necesitas Ingresar un valor en X:
6
x= 6.0
y= 18.0

```

Fig. 8. Ejecucion 1: Ejercicio 2

Para el Caso de Prueba 2. 11

X	Y	$y = x^2/2$	
15	-	$y = x^2/2$	ENTRADA
15	112.5	$y = (15)^2/2$	1 ITERACION
6	112.5	$y = 112.5$	RESULTADO

Table 11. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 9

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
cincio2_CalcularExpresion.py"
***VAMOS A CALCULAR EL VALOR DE Y EN LA SIGUIENTE EXPRESION: y=x^2/2 ***
Para lo cual necesitas Ingresar un valor en X:
15
x= 15.0
y= 112.5

```

Fig. 9. Ejecucion 2: Ejercicio 2

Para el Caso de Prueba 3. 12

X	Y	$y = x^2/2$	
4.2	-	$y = x^2/2$	ENTRADA
4.2	8.82	$y = (4.2)^2/2$	1 ITERACION
4.2	8.82	$y = 8.82$	RESULTADO

Table 12. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 10

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
cincio2_CalcularExpresion.py"
***VAMOS A CALCULAR EL VALOR DE Y EN LA SIGUIENTE EXPRESION: y=x^2/2 ***
Para lo cual necesitas Ingresar un valor en X:
4.2
x= 4.2
y= 8.82

```

Fig. 10. Ejecucion 3: Ejercicio 2

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos números entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.3 Ejercicio 3: Calculo del Área de un circulo

Dado el Radio (r) de un circulo cualquiera, calcular el valor del Área de la circunferencia que se puede construir con dicho radio.

Comprensión del problema: El ejercicio que vamos a tratar a continuación, supone ser un problema simple correspondiente al Área de la Geometría específicamente al cálculo de datos de figuras geométricas. El área de una figura es un dato correspondiente al análisis geométrico y se define como el valor de la superficie plana de un cuerpo geométrico, dicho valor puede ser calculado mediante la aplicación fórmulas matemáticas y varios datos que podemos obtener de las figuras geométricas que deseamos operar.

Para [1] los elementos de la circunferencia que se usan generalmente para resolver problemas prácticos son: diámetro, radio, arco y cuerda. Sin olvidar a una constante sumamente importante que relaciona algunos de los elementos mencionados con el círculo, estamos hablando del valor de PI equivalente a un valor aproximado de 3.1416 el cual tendrá una gran repercusión en los cálculos de Áreas que realizaremos.

De los elementos del circulo ya mencionados y dado a nuestro análisis y el objetivo de nuestro problema, el o los que serán imprescindibles para nuestro cálculo se reducen tan solo a uno. El problema planteado dicta que requerimos calcular el área de un circulo conociendo tan solo el valor de su radio, y para beneficio del enunciado el análisis geométrico nos deduce una formula con la que podemos simplificar el Análisis a tan solo una línea y utilizando el valor del que ya hablamos (radio).

Formular un modelo: Para formar un modelo algorítmico que desarrolle la operación del cálculo geométrico en cuestión partimos del análisis realizado anteriormente y nos fijamos en el o los datos que requerimos y los que buscamos.
Véase 11

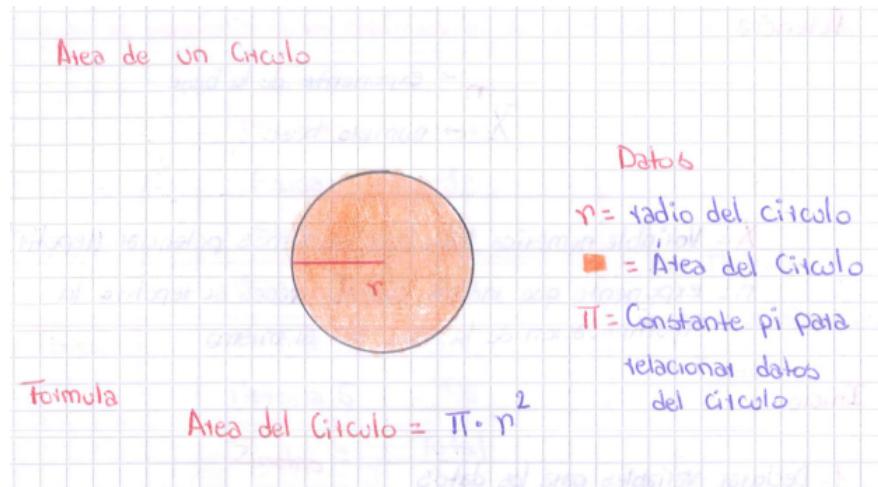


Fig. 11. Análisis para el calculo del area de un circulo

Una vez desarrollado el Análisis para el funcionamiento de nuestro programa podríamos pensar en un camino por el cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para calcular el Área de un círculo partiendo únicamente de su radio, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

Notas al texto escritas a mano

Inicio

- 1- Declarar variables para los datos
radio area PI = 3,1416
- 2- Inicio y descripción del programa
- 3- Ingreso de datos con validación
El radio debe ser un Real mayor que 0
- 3.1- El ingreso se almacena en radio
- 4- Se plantea la fórmula Area = PI * radio²
- 4.1- De acuerdo a los datos ingresados se calcula el Area y se almacena el dato
- 5- Se imprime en pantalla el dato almacenado en Area

Fin

Fig. 12. Pseudocódigo del Proceso para obtener el área de un círculo a partir del valor de su radio

Escribe el programa:

En el Listing 1.3. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular el valor del área de un círculo a partir de un radio (r) ingresado.

```

1
2 """
3 Descripción:
4 Ejercicio 3.
5 Calcular el Área de un círculo conociendo su Radio
6 Autor:
7 Erick Sebastian Mora Lara
8 """
9
10 def ingresarDatos():
11     print("**VAMOS A CALCULAR EL AREA DE UN CIRCULO A_"
12         "PARTIR_DE_SU_RADIO_(r)**")
13     print("Para comenzar Ingrese el valor del radio : ")
14     radioC = validarNumeros()
15     return radioC
16
17 def validarNumeros():
18     """
19     Es un procedimiento genérico alternativo para realizar
20     validación de números se lee un tipo de

```

```

19      dato ingresado y se intenta su conversion a flotante (
20          float), si es numero se transforman y pasa a la
21          validacion para verificar que el numero sea mayor que
22          0.
21      Si la conversion falla , se emplea el uso de
22          Excepciones para imprimir un mensaje de Error
22
23      No Recibe parametros de entrada
24
25      Retorna :
26
27          Retorna un valor entero (numero) que es el dato
28              ingresado transformado a (int)
28
29      while True:
30          num = input()
31          try:
32              num = float(num)
33              if num >0:
34                  return num
35              else:
36                  print("Solo numeros mayores a 0")
37          except ValueError:
38              print("Solo se aceptan Reales")
39
40      def areaCirculo():
41          PI = 3.1416
42          radioC = ingresarDatos()
43          areaC = PI * radioC**2
44          print(areaC)
45
46      if __name__ == '__main__':
47          areaCirculo()

```

Listing 1.3. Desarollo del ejercicio 3

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Radio	PI	Area	
4	3.1416	$3.1416 * (r)^2$	ENTRADA
4	3.1416	$3.1416 * (4)^2$	1 ITERACION
4	3.1416	50.26	RESULTADO

Table 13. Caso de Prueba 1

Radio	PI	Area	
7.5	3.1416	$3.1416 * (r)^2$	ENTRADA
7.5	3.1416	$3.1416 * (7.5)^2$	1 ITERACION
7.5	3.1416	176.71	RESULTADO

Table 14. Caso de Prueba 2

Radio	PI	Area	
2	3.1416	$3.1416 * (r)^2$	ENTRADA
2	3.1416	$3.1416 * (2)^2$	1 ITERACION
2	3.1416	12.56	RESULTADO

Table 15. Caso de Prueba 3

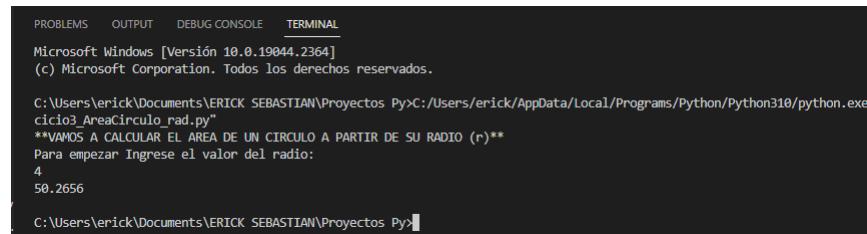
Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 16

Radio	PI	Area	
4	3.1416	$3.1416 * (r)^2$	ENTRADA
4	3.1416	$3.1416 * (4)^2$	1 ITERACION
4	3.1416	50.26	RESULTADO

Table 16. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 13



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo3_AreaCirculo_rad.py"
**VAMOS A CALCULAR EL AREA DE UN CIRCULO A PARTIR DE SU RADIO (r)**
Para empezar Ingrese el valor del radio:
4
50.2696
C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 13. Ejecucion 1: Ejercicio 3

Para el Caso de Prueba 2. 17

Radio	PI	Area	
7.5	3.1416	$3.1416 * (r)^2$	ENTRADA
7.5	3.1416	$3.1416 * (7.5)^2$	1 ITERACION
7.5	3.1416	176.71	RESULTADO

Table 17. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 14

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo3_AreaCirculo_rad.py"
**VAMOS A CALCULAR EL AREA DE UN CIRCULO A PARTIR DE SU RADIO (r)**
Para empezar Ingrese el valor del radio:
7.5
176.715

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 14. Ejecucion 2: Ejercicio 3

Para el Caso de Prueba 3. 18

Radio	PI	Area	
2	3.1416	$3.1416 * (r)^2$	ENTRADA
2	3.1416	$3.1416 * (2)^2$	1 ITERACION
2	3.1416	12.56	RESULTADO

Table 18. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 15

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo3_AreaCirculo_rad.py"
**VAMOS A CALCULAR EL AREA DE UN CIRCULO A PARTIR DE SU RADIO (r)**
Para empezar Ingrese el valor del radio:
2
12.5664

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 15. Ejecucion 3: Ejercicio 3

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.4 Ejercicio 4: Potenciación

Dado un numero base (x). Hallar la (n) potencia de la base ingresando estos dos valores.

Comprensión del problema: Se trata de un problema aritmético simple correspondiente a la potenciación como su propio nombre lo indica, lo que implica que tendremos dos valores en la operación. Para este caso vamos a denotar a un número x como la base de la operación, y otro número n que indicara el grado del exponente al que elevaremos la base. Este último indicara en términos simples cuantas veces tendrá que multiplicarse la base por sí misma.

Para este problema no se considera ninguna restricción posible, se puede hallar cualquier potencia de cualquier base escrita, por lo que el rango de los números para la operación tanto para la base como para el exponente supondrá ser los Números Reales.

Formular un modelo: El formular un modelo matemático genérico que realice cálculos de potenciación supone ser una tarea sencilla, hemos de considerar las variables antes mencionadas que corresponden tanto a un número base como a un exponente. Para propósitos del Análisis denotaremos al número base como (x) y al exponente como (n), dicho esto la notación para la potenciación se escribiría como (n^x). Donde en términos simples diríamos que la base n se multiplicara por si misma por n veces y esta expresión se leería como: “ X esta elevado a la N potencia” . Véase 16

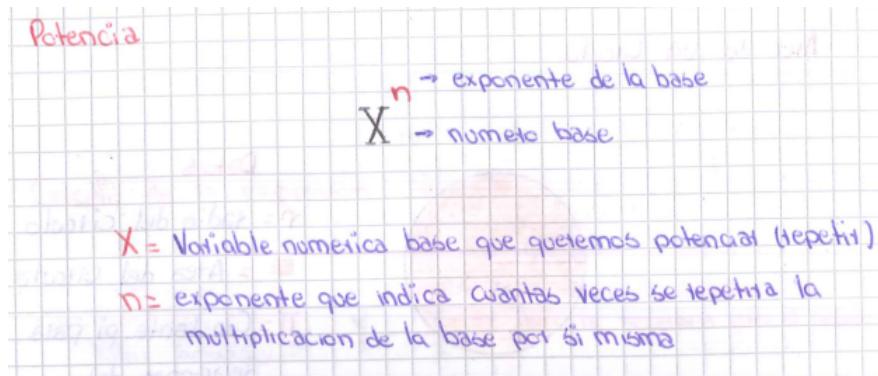


Fig. 16. Análisis para el cálculo de una potencia generica

Una vez desarrollado el Análisis que supondría un camino para el funcionamiento de nuestro programa podríamos pensar en una vía por la cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para calcular la potencia n de un número x ingresado, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

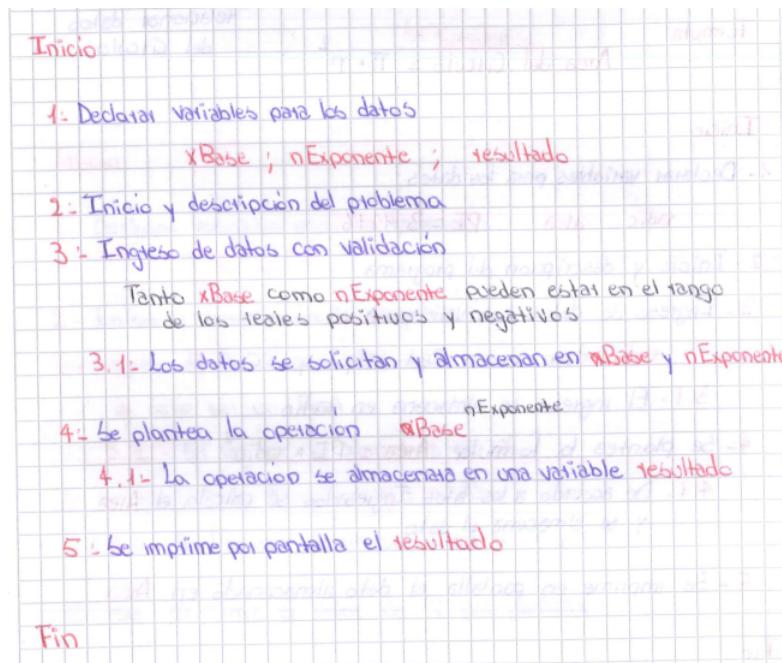


Fig. 17. Pseudocódigo del Proceso para calcular una potencia generica con X base y N exponente

Escribe el programa:

En el Listing 1.4. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular el valor de la N potencia ingresada de una base X ingresada.

```

1
2      """
3      Ejercicio 4
4      Hallar la potencia n de un numero X ingresado
5

```

```

6  Autor:
7  Erick Sebastian Mora
8  """
9  def validarNumeros():
10     """
11         Es un procedimiento generico alternativo para realizar
12             validacion de numeros enteros
13             los datos ingresados se leen como string (str) y si
14                 son numeros se verifican
15                 y se transforman en tipo de datos entero (int), en
16                     este variante de funcion
17                     se maneja el uso de una excepcion para el ingreso de
18                         datos que no son numericos:
19
20             No Recibe parametros de entrada
21
22             Retorna:
23
24             Retorna un valor entero (numero) que es el dato
25                 ingresado transformado a (int)
26
27             while True:
28                 num = input()
29                 try:
30                     num = float(num)
31                     return num
32                 except ValueError:
33                     print("Solo se aceptan Reales")
34
35             def ingresarDatos():
36                 print("Ingrese el numero base para calcular su "
37                     "potencia:")
38                 numBase = validarNumeros()
39                 print("Ingrese el numero al que desea elevar su base:")
40                 numExponente = validarNumeros()
41                 #Llamado a la operacion para calcular potencia
42                 calcularPotencia(numBase, numExponente)
43
44             def calcularPotencia(numBase, numExponente):
45                 resPotencia = numBase**numExponente
46                 print("El resultado de ", numBase, " elevado a la ",
47                     numExponente, " potencia es: ", round(resPotencia,
48                     3))
49
50             if __name__ == '__main__':
51                 ingresarDatos()

```

Listing 1.4. Desarollo del ejercicio 4

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Base(x)	Exponente(n)	Potencia	
3	2	X^n	ENTRADA
3	2	3^2	1 ITERACION
3	2	9	RESULTADO

Table 19. Caso de Prueba 1

Base(x)	Exponente(n)	Potencia	
6.2	3	X^n	ENTRADA
6.2	3	$(6.2)^3$	1 ITERACION
6.2	3	238.32	RESULTADO

Table 20. Caso de Prueba 2

Base(x)	Exponente(n)	Potencia	
9	0.5	X^n	ENTRADA
9	0.5	$9^{(0,5)}$	1 ITERACION
9	0.5	3	RESULTADO

Table 21. Caso de Prueba 3

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 22

Base(x)	Exponente(n)	Potencia	
3	2	X^n	ENTRADA
3	2	3^2	1 ITERACION
3	2	9	RESULTADO

Table 22. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 18

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ejercicio_4_Potenciacion.py"
Ingrese el numero base para calcular su potencia:
3
Ingrese el numero al que desea elevar su base:
2
El resultado de 3.0 elevado a la 2.0 potencia es: 9.0
C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 18. Ejecucion 1: Ejercicio 4

Para el Caso de Prueba 2. 23

Base(x)	Exponente(n)	Potencia	
6.2	3	X^n	ENTRADA
6.2	3	$(6.2)^3$	1 ITERACION
6.2	3	238.32	RESULTADO

Table 23. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 19

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_4_Potenciacion.py"
Ingrese el numero base para calcular su potencia:
6.2
Ingrese el numero al que desea elevar su base:
3
El resultado de 6.2 elevado a la 3.0 potencia es: 238.328

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 19. Ejecucion 2: Ejercicio 4

Para el Caso de Prueba 3. 24

Base(x)	Exponente(n)	Potencia	
9	0.5	X^n	ENTRADA
9	0.5	$9^{(0,5)}$	1 ITERACION
9	0.5	3	RESULTADO

Table 24. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 20

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_4_Potenciacion.py"
Ingrese el numero base para calcular su potencia:
9
Ingrese el numero al que desea elevar su base:
0.5
El resultado de 9.0 elevado a la 0.5 potencia es: 3.0

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 20. Ejecucion 3: Ejercicio 4

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con

los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.5 Ejercicio 5: Calculo de Notas para 5 Materias

Dadas 5 Materias realizar el Ingreso de 5 notas correspondientes a estas para posteriormente realizar los cálculos de: Suma total, promedio y Porcentaje..

Comprensión del problema: El ejercicio planteado supone ser un problema común en el ámbito educativo en el conteo de calificaciones, la estructura de resolución del problema hace uso de conceptos básicos de estadística simple como la media aritmética o promedio y que a su vez estos conceptos emplean operaciones matemáticas básicas, tales como la suma, multiplicación y división. El problema nos plantea como objetivo realizar el ingreso y conteo de 5 calificaciones correspondientes al mismo número de materias (cada nota estará relacionada a una materia) y con ello realizar una serie de cálculos solicitados como el ya mencionado promedio, la sumatoria total de las 5 calificaciones y un valor referencial que indique el porcentaje de la nota obtenida con respecto al total. Si observamos con atención cada uno de los 3 cálculos solicitados podremos evidenciar que cada uno depende de que otro sea resuelto para poder emplear su resultado en el cálculo y por ende podríamos clasificar estos cálculos en diferentes niveles de acuerdo a como se desarrolla el problema. De esta forma podríamos deducir desde el cálculo más simple que supondría ser aquel que solo dependa de los valores ingresados por teclado (calificaciones) y podemos ir escalando en complejidad hasta el cálculo que dependa de que todos hayan sido resueltos con anterioridad para poder ser aplicado. Por lo tanto al deducir el orden de los cálculos ubicamos en primera instancia la suma total de las calificaciones, puesto que con ella se podrá proceder a encontrar el promedio dividiendo el total entre la constante de 5 notas y del resultado de este cálculo podremos hacer una operación para encontrar el porcentaje de nuestro promedio obtenido en referencia al promedio máximo. Para esto supondremos en este ejercicio que la base de calificaciones esta entre 0 y 20.

Formular un modelo: Después de haber comprendido el problema el siguiente paso será plantear una forma para adaptar los conceptos revisados en el la sección anterior con el enunciado de nuestro problema. En primer lugar necesitamos contextualizarnos con el entorno del problema, necesitamos trabajar con materias y darle una respectiva calificación a cada una por lo cual entonces como primer dato tendríamos un conjunto de materias con diferentes nombres para empezar a darle forma a nuestro ejercicio. Evidentemente a cada materia del conjunto le debería corresponder una calificación y con ello entraría en juego nuestro segundo dato: El conjunto de calificaciones. Y es a partir de aquí donde

tenemos todos los datos necesarios para jugar con las distintas operaciones que hemos planteado en el Análisis. Véase 21 , 22

Calcular la sumatoria de las calificaciones de 5 materias, promedio y porcentaje

Materias	Notas
Física	17
EDO	18
Materia 3	n ₃
Materia 4	n ₄
Materia 5	n ₅
Σ Notas =	Total

5 { n Materias } { } Las Materias y Notas se almacenan en listas

$\text{Promedio} = \frac{\text{Total}}{n \text{ Materias}}$

Fig. 21. Análisis para el cálculo de 3 operaciones de 5 calificaciones

Para el porcentaje se considera que la nota máxima es 20 = 100%.

Por tanto se calcula el porcentaje con:

$$\text{Porcentaje} = \frac{\text{Promedio}}{20} \times 100\%$$

Listas

[Materia 1, Materia 2, Materia 3, Materia 4, Materia 5]
 ↓ ↓ ↓ ↓ ↓
 [Nota 1, Nota 2, Nota 3, Nota 4, Nota 5]
 Total = 0
 Resultado Total = Total + Nota [1,2,3,4,5]
 Sumatoria

Fig. 22. Análisis para el cálculo de 3 operaciones de 5 calificaciones

Una vez desarrollado el Análisis que supondría un camino para el funcionamiento de nuestro programa podríamos pensar en una vía por la cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para calcular la Suma, el promedio y el porcentaje de las calificaciones ingresadas de 5 materias, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

Inicio

1- Declarar variables para los Datos
materias, notas, listaMaterias ; listaNotas ; totalNotas
promedio ; porcentaje

2 - Inicio y descripción del problema

3 - Ingresar con un ciclo repetitivo For un total de 5 materias
cada una se leerá como una cadena de caracteres y se
almacena en una lista vacía listaMaterias

3.1 Regresar la lista con las 5 materias agregadas

4 - Con la lista de Materias creada iniciaremos un ciclo For
que se repite por cada elemento de la lista solicitando
en Valores Reales del 0 al 20 cada una de las notas
y las almacene en la lista vacía listaNotas

4.1 - Regresar la lista con las 5 notas agregadas

5.- Con la lista de Notas `listaNotas` plantea las operaciones

5.1- Planteamos un ciclo `For` que se repita por cada elemento de la `listaNotas` y que extraiga cada uno de los valores para sumarlos al resultado que fue inicializado con 0

$$\text{totalNotas} = \text{totalNotas} + \text{Nota}$$

5.1.1- Imprimimos el valor `totalNotas` en pantalla

5.2- Con el valor `totalNotas` podemos calcular el promedio
conociendo que tenemos 5 materias

$$\text{promedio} = \frac{\text{totalNotas}}{5}$$

5.2.1- Imprimimos por pantalla el valor `promedio`

5.3- Con el valor `promedio` podemos calcular el porcentaje
planteando la formula

$$\text{Porcentaje} = \frac{\text{Promedio}}{20} \times 100\%$$

5.3.1- Imprimimos por pantalla el valor `porcentaje + "%"`

Fin

Fig. 23. Pseudocódigo del Proceso para calcular 3 operaciones a partir de 5 calificaciones

Escribe el programa:

En el Listing 1.5. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular el valor total, el promedio y el porcentaje referencial de las calificaciones de 5 materias

```

1
2      """
3  Descripción:
4  Ejercicio 5.
5  Ingresar las calificaciones de 5 materias, y calcular el
    total, el promedio y porcentaje

```

```

6   Autor:
7   Erick Sebastian Mora
8   """
9
10  def validarNumeros():
11      """
12          Es un procedimiento generico alternativo para realizar
13          validacion de numeros se lee un tipo de
14          dato ingresado y se intenta su conversion a flotante (
15              float), si es numero se transforman y pasa a la
16          validacion para verificar que el numero sea mayor que
17              0.
18          Si la conversion falla , se emplea el uso de
19              Excepciones para imprimir un mensaje de Error
20
21      No Recibe parametros de entrada
22
23      Retorna:
24
25          Retorna un valor entero (numero) que es el dato
26          ingresado transformado a (int)
27      """
28
29      while True:
30          num = input()
31          try:
32              num = float(num)
33              if num >=0 and num<21:
34                  return num
35              else:
36                  print("Las notas deben ser entre 0 y 20")
37          except ValueError:
38              print("Solo se aceptan Reales")
39
40  def ingresarMaterias():
41      listaMaterias = []
42      for i in range(5):
43          print("Ingrese el nombre de la Materia ", i+1, ":")
44          materias = input()
45          listaMaterias.append(materias)
46      return listaMaterias
47
48  def ingresarCalificaciones():
49      listaMaterias = ingresarMaterias()
50      listaCalificaciones = []
51      for materia in listaMaterias:
52          print("Ingrese la calificacion de ", materia, ":")
53          nota = validarNumeros()
54          listaCalificaciones.append(nota)

```

```

49     return listaCalificaciones
50
51 def totalCalificaciones():
52     totalNotas = 0
53     promedioNotas = 0
54     porcentNotas = 0
55     listaCalificaciones = ingresarCalificaciones()
56
57     for notas in listaCalificaciones:
58         totalNotas = totalNotas + notas
59
60     print("La Sumatoria de las Notas Ingresadas es: " ,
61           totalNotas)
61     promedioNotas=totalNotas/5
62     print("El Promedio de las Notas es: " , promedioNotas)
63     porcentNotas = (promedioNotas*100)/20
64     print("El porcentaje de las notas es del: " ,
65           porcentNotas , "%")
66
67 if __name__ == '__main__':
68     totalCalificaciones()

```

Listing 1.5. Desarrollo del ejercicio 5

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-Mora_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Materias	Notas	Sumatoria	Promedio	Porcentaje	
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	-	-	-	ENTRADA
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	17+18+17+19+18	-	-	1 ITERACION
[EDO, Química, Modelos, Ingles, POO]	[17,18,17,19,18]	89	89/5	-	2 ITERACION
[EDO, Química, Modelos, Ingles, POO]	[17,18,17,19,18]	89	17.8	(17.8*100)/20	3 ITERACION
[EDO, Química, Modelos, Ingles, POO]	[17,18,17,19,18]	89	17.8	89%	RESULTADO

Table 25. Caso de Prueba 1

Materias	Notas	Sumatoria	Promedio	Porcentaje	
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	-	-	-	ENTRADA
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	19+20+15+16+17	-	-	1 ITERACION
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	87	87/5	-	2 ITERACION
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	87	17.4	(17.4*100)/20	3 ITERACION
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	87	17.4	87%	RESULTADO

Table 26. Caso de Prueba 2

Materias	Notas	Sumatoria	Promedio	Porcentaje	
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	-	-	-	ENTRADA
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	20+16+15+13+16	-	-	1 ITERACION
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	80	80/5	-	2 ITERACION
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	80	16	(16*100)/20	3 ITERACION
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	80	16	80%	RESULTADO

Table 27. Caso de Prueba 3

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 28

Materias	Notas	Sumatoria	Promedio	Porcentaje	
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	-	-	-	ENTRADA
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	17+18+17+19+18	-	-	1 ITERACION
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	89	89/5	-	2 ITERACION
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	89	17.8	(17.8*100)/20	3 ITERACION
[EDO, Quimica, Modelos, Ingles, POO]	[17,18,17,19,18]	89	17.8	89%	RESULTADO

Table 28. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 24

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
/ERICK SEBASTIAN/Proyectos Py/Ejercicio_5_Materias.py"
Ingrese el nombre de la Materia 1 :
EDO
Ingrese el nombre de la Materia 2 :
Quimica
Ingrese el nombre de la Materia 3 :
Modelos
Ingrese el nombre de la Materia 4 :
Ingles
Ingrese el nombre de la Materia 5 :
POO
Ingrese la calificacion de EDO :
17
Ingrese la calificacion de Quimica :
18
Ingrese la calificacion de Modelos :
17
Ingrese la calificacion de Ingles :
19
Ingrese la calificacion de POO :
18
La Sumatoria de las Notas Ingresadas es: 89.0
El Promedio de las Notas es: 17.8
El porcentaje de las notas es del: 89.0 %

```

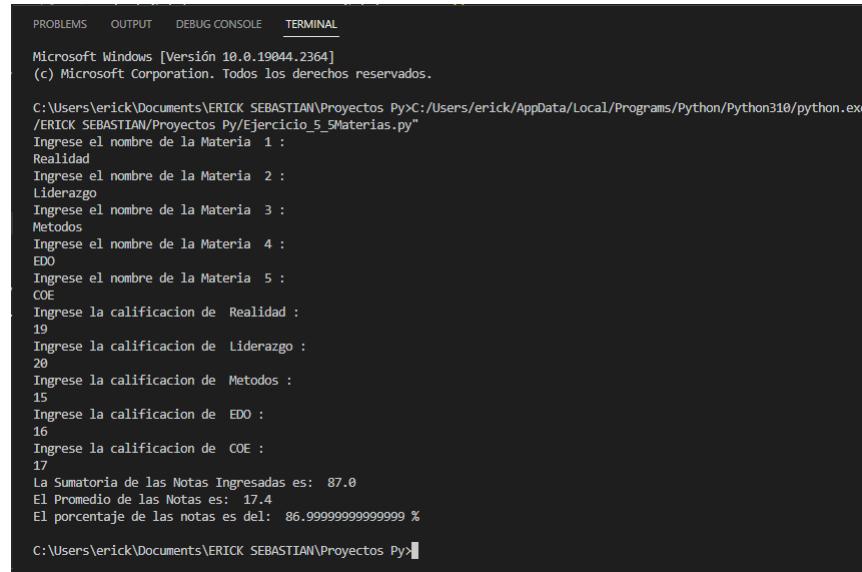
Fig. 24. Ejecucion 1: Ejercicio 5

Para el Caso de Prueba 2. 29

Materias	Notas	Sumatoria	Promedio	Porcentaje	
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	-	-	-	ENTRADA
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	19+20+15+16+17	-	-	1 ITERACION
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	87	87/5	-	2 ITERACION
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	87	17.4	(17.4*100)/20	3 ITERACION
[Realidad, Liderazgo, Metodos, EDO, COE]	[19,20,15,16,17]	87	17.4	87%	RESULTADO

Table 29. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 25



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
/ERICK SEBASTIAN/Proyectos Py/Ejercicio_5_Materias.py"
Ingrese el nombre de la Materia 1 :
Realidad
Ingrese el nombre de la Materia 2 :
Liderazgo
Ingrese el nombre de la Materia 3 :
Metodos
Ingrese el nombre de la Materia 4 :
EDO
Ingrese el nombre de la Materia 5 :
COE
Ingrese la calificacion de Realidad :
19
Ingrese la calificacion de Liderazgo :
20
Ingrese la calificacion de Metodos :
15
Ingrese la calificacion de EDO :
16
Ingrese la calificacion de COE :
17
La Sumatoria de las Notas Ingresadas es: 87.0
El Promedio de las Notas es: 17.4
El porcentaje de las notas es del: 86.99999999999999 %

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 25. Ejecucion 2: Ejercicio 5

Para el Caso de Prueba 3. 30

Materias	Notas	Sumatoria	Promedio	Porcentaje	
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	-	-	-	ENTRADA
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	20+16+15+13+16	-	-	1 ITERACION
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	80	80/5	-	2 ITERACION
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	80	16	(16*100)/20	3 ITERACION
[Lengua, Historia, COE, POO, Fisica]	[20,16,15,13,16]	80	16	80%	RESULTADO

Table 30. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 26

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
/ERICK SEBASTIAN/Proyectos Py/Ejercicio_5_Materias.py"
Ingrese el nombre de la Materia 1 :
Lengua
Ingrese el nombre de la Materia 2 :
Historia
Ingrese el nombre de la Materia 3 :
COE
Ingrese el nombre de la Materia 4 :
POO
Ingrese el nombre de la Materia 5 :
Fisica
Ingrese la calificacion de Lengua :
28
Ingrese la calificacion de Historia :
16
Ingrese la calificacion de COE :
15
Ingrese la calificacion de POO :
13
Ingrese la calificacion de Fisica :
16
La Sumatoria de las Notas Ingresadas es: 80.0
El Promedio de las Notas es: 16.0
El porcentaje de las notas es del: 80.0 %

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 26. Ejecucion 3: Ejercicio 5

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.6 Ejercicio 6: Teorema de Pitágoras

Calcular la hipotenusa de un triángulo rectángulo mediante la función del Teorema de pitágoras: $h^2 = a^2 + b^2$.

Comprensión del problema: El problema corresponde a un ejercicio con aplicación de fundamentos de geometría y matemática, explícitamente se nos presenta el caso del Teorema de Pitágoras para resolver un lado desconocido de un triángulo rectángulo partiendo de sus dos lados restantes, el cálculo se basa en usar los dos lados conocidos del triángulo y emplearlos en la fórmula del Teorema de Pitágoras para encontrar su lado faltante.

De este modo se nos pueden presentar dos casos distintos para el mismo problema y tendremos que diferenciar y tomar en cuenta en qué caso nos hallamos según ciertos parámetros dados. En primera instancia un caso supone un triángulo

rectángulo donde los dos lados conocidos son dos catetos (lados más pequeños del triángulo) mientras que el caso restante propone al triángulo rectángulo con un cateto y la hipotenusa como lados conocidos (la hipotenusa es el lado más largo del triángulo).

Ambos casos pueden parecer significativamente similares pero para efectos del cálculo suponen una diferencia notable, y es que si partimos del Teorema de Pitágoras $h^2 = a^2 + b^2$: Donde h representa a la hipotenusa y tanto a como b suponen ser ambos catetos del triángulo, debemos tener en cuenta que la hipotenusa siempre será significativamente mayor que ambos catetos o de lo contrario no estaríamos hablando de un triángulo rectángulo al cual le podemos aplicar el Teorema de Pitágoras para hallar uno de sus lados.

Esto se evidencia si queremos hallar uno de los catetos partiendo de la hipotenusa conocida junto a otro cateto, dada una suposición de que la hipotenusa sea un valor menor al cateto conocido el cálculo recaería en una raíz negativa en la que es imposible tanto matemáticamente como para las definiciones geométricas tener un valor imaginario como medida.

Formular un modelo: Tras comprender el problema y haber desarrollado un análisis pertinente podemos proponer el Teorema de Pitágoras en sus dos casos para relacionarlo con nuestro programa genérico.

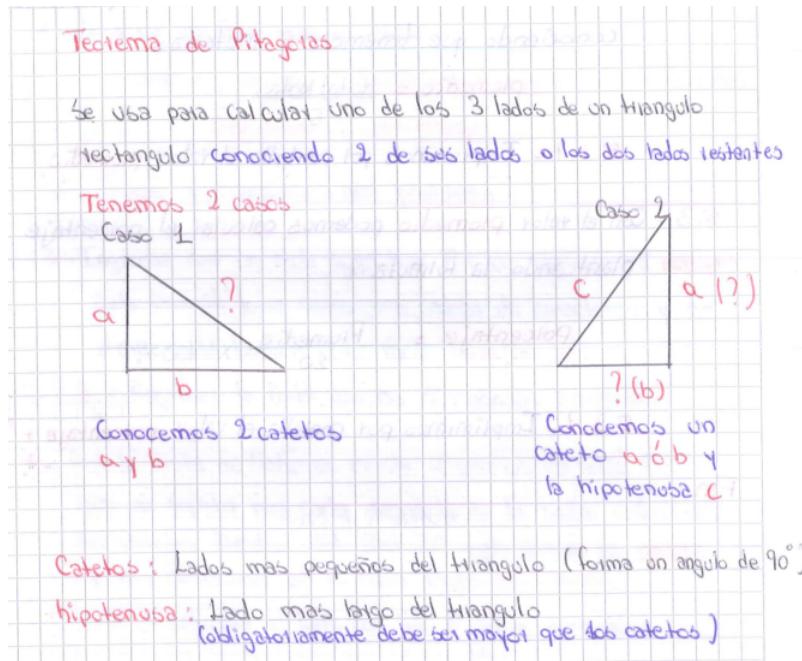
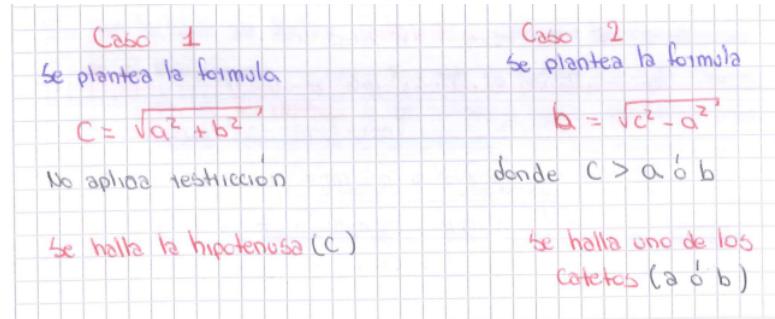


Fig. 27. Análisis para el cálculo genérico del Teorema de Pitágoras

**Fig. 28.** Analisis para el calculo generico del Teorema de Pitagoras

Una vez desarrollado el Análisis que supondria un camino para el funcionamiento de nuestro programa podríamos pensar en una vía por la cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para calcular el lado restante de un triángulo rectángulo utilizando el teorema de pitágoras, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

Inicio

1- Declarar variables para los Datos

materias, notas, listaMaterias ; listaNotas ; totalNotas
promedio ; porcentaje

2 - Inicio y descripción del problema

3- Ingresar con un ciclo repetitivo For un total de 5 materias
cada una se leera como una cadena de caracteres y se
almacenaria en una lista vacia listaMaterias

3.1 Regresar la lista con las 5 materias agregadas

4- Con la lista de Materias creada iniciaremos un ciclo For
que se repite por cada elemento de la lista solicitando
en Valores Reales del 0 al 20 cada una de las notas
y las almacene en la lista vacia listaNotas

4.1 - Regresar la lista con las 5 notas agregadas

4.2- Caso 2 cateto e hipotenusa

4.2.1- Ingreso de datos con validación

Cateto e hipotenusa deben ser reales positivos y
la hipotenusa debe ser mayor que cateto

4.2.2 - Planteamos la formula

$$\text{Cateto}_x = \sqrt{(\text{hipotenusa})^2 - (\text{cateto}_y)^2}$$

4.2.3 - Almacenamos Cateto_x e imprimimos
por pantalla

5- Para el ultimo caso definimos la salida del programa
FIN

Fig. 29. Pseudocodigo del Proceso para hallar los lados de un triangulo rectangulo mediante el teorema de Pitagoras

Escribe el programa:

En el Listing 1.6. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular el lado faltante de un triangulo rectangulo utilizando el teorema de Pitagoras.

```

1
2           """
3   Descripcion :
4   Ejercicio 6.
5   Calcular la hipotenusa de un triangulo rectangulo mediante
       la funcion de pitagoras
6    $h^2 = a^2 + b^2$ 
7   Autor:
8   Erick Sebastian Mora Lara
9   """
10
11  def validarNumeros():
12      """
13          Es un procedimiento generico alternativo para realizar
              validacion de numeros enteros
14          los datos ingresados se leen como string (str) y si
              son numeros se verifican
15          y se transforman en tipo de datos entero (int), en
              este variante de funcion
16          se maneja el uso de una excepcion para el ingreso de
              datos que no son numericos:
17
18          No Recibe parametros de entrada
19
20      Retorna:
21
22          Retorna un valor entero (numero) que es el dato
              ingresado transformado a (int)
23      """
24      while True:
25          num = input()
26          try:
27              num = float(num)
28              if num >0:
29                  return num

```

```

30
31         else:
32             print("Solo numeros mayores a 0")
33     except ValueError:
34         print("Solo se aceptan Reales")
35
36 def menuOpciones():
37     print("Vamos a calcular un lado del triangulo "
38         "usando la funcion de pitagoras")
39     print("Necesitamos conocer al menos dos lados del "
40         "triangulo:")
41     print("1.- 2.Catetos")
42     print("2.- Hipotenusa y Cateto")
43     print("3.- Salir")
44     opcion = input("Ingrese el numero de la opcion "
45         "deseada:")
46     return opcion
47
48 def seleccionOpcion(opcion):
49     while True:
50         if opcion == "1":
51             dosCatetos()
52         elif opcion == "2":
53             catetoHipotenusa()
54         elif opcion == "3":
55             salir()
56         else:
57             print("Opcion invalida. Intentelo de nuevo.")
58
59 def dosCatetos():
60     print("Ingresa el valor del Cateto Uno")
61     catetoUno = validarNumeros()
62     print("Ingresa el valor del Cateto Dos")
63     catetoDos = validarNumeros()
64
65     resHipotenusa = (catetoUno**2 + catetoDos**2)**0.5
66
67     print("El valor de la Hipotenusa es:", resHipotenusa)
68
69 def catetoHipotenusa():
70     hipotenusa = 0
71     catetoUno = 0
72     while(catetoUno >= hipotenusa):
73         print("Ingresa el Valor de la Hipotenusa:")
74         hipotenusa = validarNumeros()
75         print("Ingresa el Valor de un Cateto:")
76         catetoUno = validarNumeros()
77         if(catetoUno >= hipotenusa):
78             print("Los catetos no pueden "
79                 "ser mas grandes que la "
80                 "hipotenusa")
81
82         else:
83             print("Los catetos cumplen con la "
84                 "condicion de Pitagoras")
85
86         print("Deseas calcular otro triangulo? S/N")
87         respuesta = input()
88         if respuesta == "N" or respuesta == "n":
89             break
90
91     print("Gracias por usar el calculadora de triangulos")
92
93     salir()

```

```

    hipotenusa , „Ingresar otro
    valor”)
75      catetoDos = (hipotenusa**2 - catetoUno**2)
76          **0.5
77
78      print(“El valor del cateto restante es : ”, catetoDos)
79
80  def salir():
81      print(“Saliendo”)
82      exit()
83
84 if __name__ == ‘__main__':
85     nuevoMenu = menuOpciones()
86     seleccionOpcion(nuevoMenu)

```

Listing 1.6. Desarrollo del ejercicio 6

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Cateto 1	Cateto 2	Hipotenusa	
3	2	-	ENTRADA
3	2	$h = \sqrt{3^2 + 1^2}$	1 ITERACION
3	2	$h = 3.6$	RESULTADO

Table 31. Caso de Prueba 1

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Cateto a	Cateto b	Hipotenusa	
-	6	9	ENTRADA
$a = \sqrt{9^2 - 6^2}$	6	9	1 ITERACION
$a = 6.7$	6	9	RESULTADO

Table 32. Caso de Prueba 2

Cateto a	Cateto b	Hipotenusa	
4	-	10	ENTRADA
4	$b = \sqrt{10^2 - 4^2}$	10	1 ITERACION
4	$b = 9.16$	10	RESULTADO

Table 33. Caso de Prueba 3

Para el Caso de Prueba 1. 34

Cateto 1	Cateto 2	Hipotenusa	
3	2	-	ENTRADA
3	2	$h = \sqrt{3^2 + 1^2}$	1 ITERACION
3	2	$h = 3.6$	RESULTADO

Table 34. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 30

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ejercicio_6_Pitagoras.py"
Vamos a calcular un lado del triangulo usando la funcion de pitagoras
Necesitamos conocer al menos dos lados del triangulo:
1.- 2 Catetos
2.- Hipotenusa y Cateto
3.- Salir
Ingrese el número de la opción deseada: 1
Ingrese el valor del Cateto Uno
3
Ingresa el valor del Cateto Dos
2
El valor de la Hipotenusa es:  3.60551275463989

```

Fig. 30. Ejecucion 1: Ejercicio 6

Para el Caso de Prueba 2. 35

Cateto a	Cateto b	Hipotenusa	
-	6	9	ENTRADA
$a = \sqrt{9^2 - 6^2}$	6	9	1 ITERACION
$a = 6.7$	6	9	RESULTADO

Table 35. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 31

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
/ERICK SEBASTIAN/Proyectos Py/Ejercicio_5_Materias.py"
Ingrese el nombre de la Materia 1 :
Realidad
Ingrese el nombre de la Materia 2 :
Liderazgo
Ingrese el nombre de la Materia 3 :
Metodos
Ingrese el nombre de la Materia 4 :
EDO
Ingrese el nombre de la Materia 5 :
COE
Ingrese la calificacion de Realidad :
19
Ingrese la calificacion de Liderazgo :
20
Ingrese la calificacion de Metodos :
15
Ingrese la calificacion de EDO :
16
Ingrese la calificacion de COE :
17
La Sumatoria de las Notas Ingresadas es: 87.0
El Promedio de las Notas es: 17.4
El porcentaje de las notas es del: 86.99999999999999 %

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

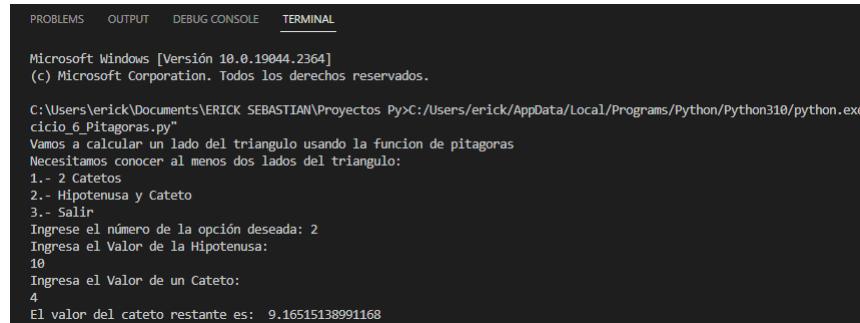
Fig. 31. Ejecucion 2: Ejercicio 6

Para el Caso de Prueba 3. 36

Cateto a	Cateto b	Hipotenusa	
4	-	10	ENTRADA
4	$b = \sqrt{10^2 - 4^2}$	10	1 ITERACION
4	$b = 9.16$	10	RESULTADO

Table 36. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 32



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ejercicio_6_Pitagoras.py"
Vamos a calcular un lado del triangulo usando la funcion de pitagoras
Necesitamos conocer al menos dos lados del triangulo:
1.- 2 Catetos
2.- Hipotenusa y Cateto
3.- Salir
Ingrese el número de la opción deseada: 2
Ingresá el Valor de la Hipotenusa:
10
Ingresá el Valor de un Cateto:
4
El valor del cateto restante es: 9.16515138991168

```

Fig. 32. Ejecucion 3: Ejercicio 6

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.7 Ejercicio 7: Conversor de Libras a Kilos y gramos

Dada una medida ingresada en Libras se propone la conversion de el valor a Kilogramos y Gramos

Comprensión del problema: El ejercicio planteado supone ser un problema de conversión de unidades que para el caso, dado un valor Real se lo tomara como 'Libras' y a partir de él se realizara la transformación a otras unidades de peso. En este caso el enunciado nos propone realizar la conversión a Kilos y a gramos respectivamente, para este proceso existen un par de fórmulas conocidas como "Factores de conversión" que simplificaran el cálculo a una operación matemática simple, y nos darán un equivalente entre las unidades en cuestión.

Formular un modelo: Hemos determinado tras el Análisis que resolver nuestro problema no implica complejidad alguna, si trabajamos con los ya mencionados factores de conversión el cálculo entre unidades se simplificaría a una sencilla multiplicación. Se desarrollaría entonces el Análisis de los factores de conversión que se emplearan en la resolución del ejercicio.

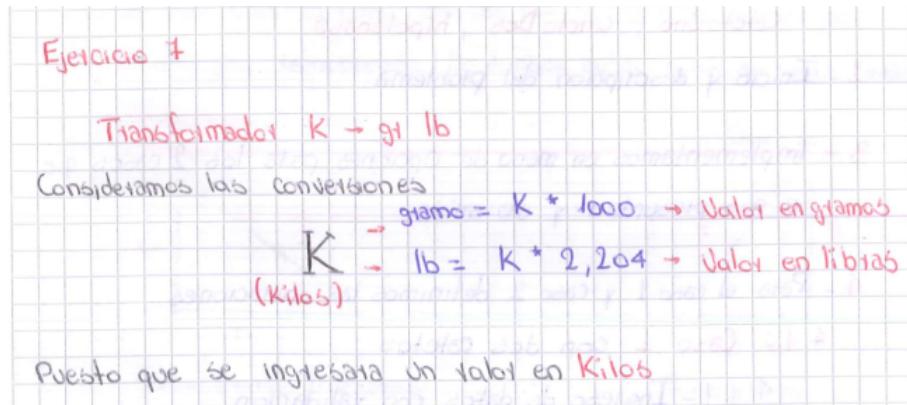


Fig. 33. Análisis para la conversión de Unidades de Peso

Una vez desarrollado el Análisis que supondría un camino para el funcionamiento de nuestro programa podríamos pensar en una vía por la cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para realizar la conversión de entre unidades de Peso utilizando Factores de Conversión, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

Inicio

- 1- Declarar variables para los datos
Kilo ; gramo ; libra
- 2- Inicio y descripción del programa
- 3- Ingreso de datos con validación (Kilo)
El Valor de Kilo debe ser mayor o igual a 0
- 4- El valor ingresado se almacena en la variable Kilo
- 5- Planteamos las operaciones
 - 5.1- Para la transformación a gramos $gt = K \times 1000 \rightarrow$ gramos
 - 5.2- Para la transformación a libras $lb = K \times 2,204 \rightarrow$ libras
 - 5.3- los datos se almacenan en las variables gramos y libras
- 6- Se imprimen por pantalla los valores de gramo y libra

Fin

Escribe el programa:

En el Listing 1.7. Se encuentra el desarrollo del algoritmo cuyo objetivo es Transformar unidades de peso entre Libras a Kilos y Libras a gramos.

```
1
2     """
3     Descripcion :
4     Ejercicio 7.
5     Transformar una medida ingresada en Libras a Kilogramos y
6     Gramos
7     Autor :
8     Erick Sebastian Mora Lara
9     """
10    def validarNumeros():
11        """
12            Es un procedimiento generico alternativo para realizar
13            la validacion de numeros enteros
14            los datos ingresados se leen como string (str) y si
15            son numeros se verifican
16            y se transforman en tipo de datos entero (int), en
17            este variante de funcion
18            se maneja el uso de una excepcion para el ingreso de
19            datos que no son numericos:
20
21            No Recibe parametros de entrada
22
```

```
19     Retorna:  
20     _____  
21         Retorna un valor entero (numero) que es el dato  
22             ingresado transformado a (int)  
23     """  
24     while True:  
25         num = input()  
26         try:  
27             num = float(num)  
28             if num >0:  
29                 return num  
30             else:  
31                 print("Solo numeros mayores a 0")  
32         except ValueError:  
33             print("Solo se aceptan Reales")  
34  
35     def ingresarDatos():  
36         print("Ingrese su peso en Libras (lb): ")  
37         peso = validarNumeros()  
38         transformarKilos(peso)  
39         transformarGramos(peso)  
40  
41     def transformarKilos(libras):  
42         kilos = libras/2.2046  
43         print(libras," libras equivalen a ",round(kilos, 4)," Kilos")  
44  
45     def transformarGramos(libras):  
46         gramos = libras*453.59  
47         print(libras," libras equivalen a ",round(gramos,4)," gramos")  
48  
49  
50     if __name__ == '__main__':  
51         ingresarDatos()
```

Listing 1.7. Desarollo del ejercicio 7

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Libra	Kilo	Gramo	
3	-	-	ENTRADA
3	$k = lb/2.204$	$gr = lb * 453.59$	1 ITERACION
3	1.3611	1360.77	RESULTADO

Table 37. Caso de Prueba 1

Libra	Kilo 2	Gramo	
7.4	-	-	ENTRADA
7.4	$k = lb/2.204$	$gr = lb * 453.59$	1 ITERACION
7.4	3.35	3356.56	RESULTADO

Table 38. Caso de Prueba 2

Libra	Kilo 2	Gramo	
14	-	-	ENTRADA
14	$k = lb/2.204$	$gr = lb * 453.59$	1 ITERACION
14	6.35	6350.26	RESULTADO

Table 39. Caso de Prueba 3

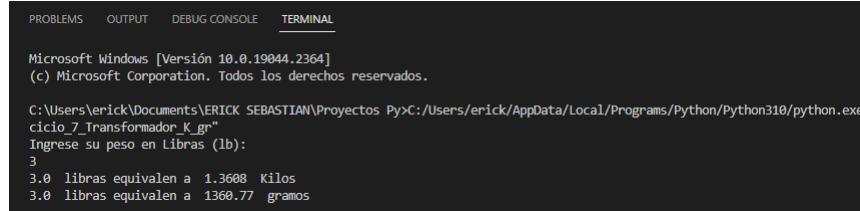
Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 40

Libra	Kilo	Gramo	
3	-	-	ENTRADA
3	$k = lb / 2.204$	$gr = lb * 453.59$	1 ITERACION
3	1.3611	1360.77	RESULTADO

Table 40. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 34



```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ejercicio_7_Transformador_K_gr"
Ingrese su peso en Libras (lb):
3
3.0 libras equivalen a 1.3608 Kilos
3.0 libras equivalen a 1360.77 gramos

```

Fig. 34. Ejecucion 1: Ejercicio 7

Para el Caso de Prueba 2. 41

Libra	Kilo 2	Gramo	
7.4	-	-	ENTRADA
7.4	$k = lb / 2.204$	$gr = lb * 453.59$	1 ITERACION
7.4	3.35	3356.56	RESULTADO

Table 41. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 35

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe ejercicio_7_Transformador_K_gr.py
Ingrese su peso en Libras (lb):
7.4
7.4 libras equivalen a 3.3566 Kilos
7.4 libras equivalen a 3356.566 gramos

```

Fig. 35. Ejecucion 2: Ejercicio 7

Para el Caso de Prueba 3. 42

Libra	Kilo	Gramo	
14	-	-	ENTRADA
14	$k = lb / 2.204$	$gr = lb * 453.59$	1 ITERACION
14	6.35	6350.26	RESULTADO

Table 42. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 36

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe ejercicio_7_Transformador_K_gr.py
Ingrese su peso en Libras (lb):
14
14.0 libras equivalen a 6.3504 Kilos
14.0 libras equivalen a 6350.26 gramos

```

Fig. 36. Ejecucion 3: Ejercicio 7

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.8 Ejercicio 8: Conversor de Dolares a Euros y Yenes

Dado un valor ingresado en Dolares se propone la conversion de este a Euros y Yenes respectivamente.

Comprendión del problema: El ejercicio planteado supone ser un problema de conversión de unidades de tipo monetarias que para el caso, dado un valor Real se lo tomara como 'Dolares' y a partir de él se realizara la transformación a otras unidades monetarias. En este caso el enunciado nos propone realizar la conversión a Euros y a Yenes respectivamente, para este proceso existen un par de fórmulas conocidas como "Factores de conversión" que simplificaran el cálculo a una operación matemática simple, y nos darán un equivalente entre las unidades en cuestión.

Formular un modelo: Hemos determinado tras el Análisis que resolver nuestro problema no implica complejidad alguna, si trabajamos con los ya mencionados factores de conversion el calculo entre unidades se simplificaria a una sencilla multiplicacion. Se desarollara entonces el Análisis de los factores de conversion que se emplearan en la resolucion del ejercicio.

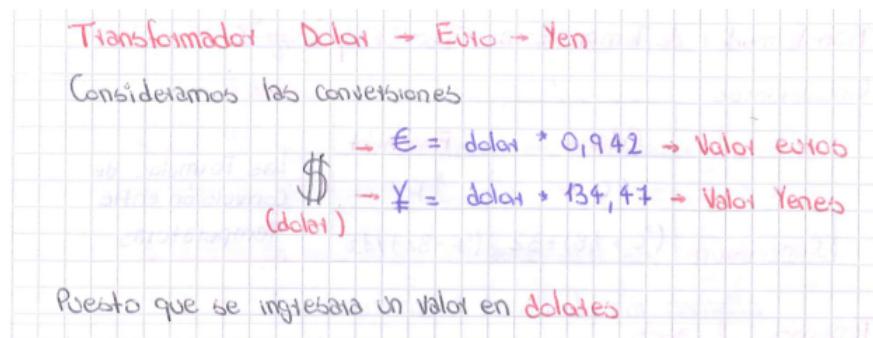
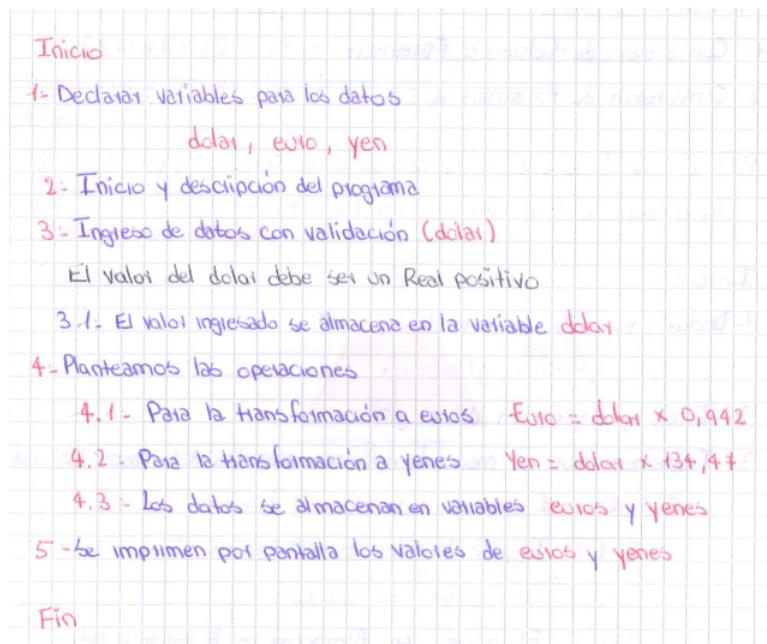


Fig. 37. Análisis para la conversion de Unidades Monetarias

Una vez desarrollado el Análisis que supondria un camino para el funcionamiento de nuestro programa podríamos pensar en una vía por la cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para realizar la Transformacion de entre unidades Monetarias utilizando Factores de Conversion, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la

solución.



Escribe el programa:

En el Listing 1.8. Se encuentra el desarrollo del algoritmo cuyo objetivo es Transformar unidades Monetarias partiendo del Dolar a Euro y Yen.

```

1
2      """
3  Descripción:
4  Ejercicio 5.
5  Ingresar un valor en dólares y transformar en euros y yen
.
6  Autor:
7  Erick Sebastian Mora
8  """
9
10 def validarNumeros():
11     """
12     Es un procedimiento generico alternativo para realizar
13     validacion de numeros enteros
14     los datos ingresados se leen como string (str) y si
15     son numeros se verifican
16     y se transforman en tipo de datos entero (int), en
17     este variante de funcion

```

```

15      se maneja el uso de una excepcion para el ingreso de
16      datos que no son numericos:
17      _____
18      No Recibe parametros de entrada
19      Retorna:
20      _____
21      Retorna un valor entero (numero) que es el dato
22      ingresado transformado a (int)
22      """
23      while True:
24          num = input()
25          try:
26              num = float(num)
27              if num >0:
28                  return num
29              else:
30                  print("Solo numeros mayores a 0")
31          except ValueError:
32              print("Solo se aceptan Reales")
33
34      def ingresarDatos():
35          print("Ingrese su valor en Dolares ($): ")
36          dolar = validarNumeros()
37          transformarEuros(dolar)
38          transformarYen(dolar)
39
40      def transformarEuros(dolar):
41          euro = dolar*0.942
42          print(dolar," dolares equivalen a ",round(euro,2)," euros")
43
44      def transformarYen(dolar):
45          yen = dolar*134.47
46          print(dolar," dolares equivalen a ",round(yen,2)," Yenes")
47
48      if __name__ == '__main__':
49          ingresarDatos()

```

Listing 1.8. Desarollo del ejercicio 8

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Dolar	Euro	Yen	
15	-	-	ENTRADA
15	$Euro = dolar * 0.942$	$Yen = dolar * 134.47$	1 ITERACION
15	14.13	2017.05	RESULTADO

Table 43. Caso de Prueba 1

Dolar	Euro	Yen	
12.5	-	-	ENTRADA
12.5	$Euro = dolar * 0.942$	$Yen = dolar * 134.47$	1 ITERACION
12.5	11.77	1680.87	RESULTADO

Table 44. Caso de Prueba 2

Dolar	Euro	Yen	
5	-	-	ENTRADA
5	$Euro = dolar * 0.942$	$Yen = dolar * 134.47$	1 ITERACION
5	4.71	672.35	RESULTADO

Table 45. Caso de Prueba 3

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 46

Dolar	Euro	Yen	
15	-	-	ENTRADA
15	$Euro = dolar * 0.942$	$Yen = dolar * 134.47$	1 ITERACION
15	14.13	2017.05	RESULTADO

Table 46. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 38

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_8_Transformador_Euro_Yen.py"
Ingrese su valor en Dolares ($):
15
15.0 dolares equivalen a 14.13 euros
15.0 dolares equivalen a 2017.05 Yenes
C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 38. Ejecucion 1: Ejercicio 8

Para el Caso de Prueba 2. 47

Dolar	Euro	Yen	
12.5	-	-	ENTRADA
12.5	$Euro = dolar * 0.942$	$Yen = dolar * 134.47$	1 ITERACION
12.5	11.77	1680.87	RESULTADO

Table 47. Caso de Prueba 2

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 39

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
cambio_8_Transformador_Euro_Yen.py"
Ingrese su valor en Dolares ($):
12.5
12.5 dolares equivalen a 11.77 euros
12.5 dolares equivalen a 1680.88 Yenes

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 39. Ejecucion 2: Ejercicio 8

Para el Caso de Prueba 3. 48

Dolar	Euro	Yen	
5	-	-	ENTRADA
5	$Euro = dolar * 0.942$	$Yen = dolar * 134.47$	1 ITERACION
5	4.71	672.35	RESULTADO

Table 48. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 40

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
cambio_8_Transformador_Euro_Yen.py"
Ingrese su valor en Dolares ($):
5
5.0 dolares equivalen a 4.71 euros
5.0 dolares equivalen a 672.35 Yenes

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 40. Ejecucion 3: Ejercicio 8

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por

concluido el ejercicio.

2.9 Ejercicio 9: Conversor de Unidades de Temperatura (Celsius - Farenheit)

Dado un valor real ingresado correspondiente a una unidad de Temperatura, se plantea realizar una conversion de Celcius a Farenheit y viceversa.

Comprendión del problema: El ejercicio planteado supone ser un problema de conversión de unidades correspondientes a Temperatura que para el caso, dado un valor Real existiran dos posibles casos. Realizar una conversion a grados Farenheit tomando el valor ingresado como unidades Celsius o hacer la operacion contraria que implicaria tomar el valor ingresado como unidades Farenheit y realizar una conversion a grados Celcius. Para este proceso existen un par de fórmulas conocidas como "Factores de conversión" que simplificaran el cálculo a dos operaciones matemáticas simples dado los dos casos presentados, y nos darán como resultado un equivalente entre las unidades en cuestión.

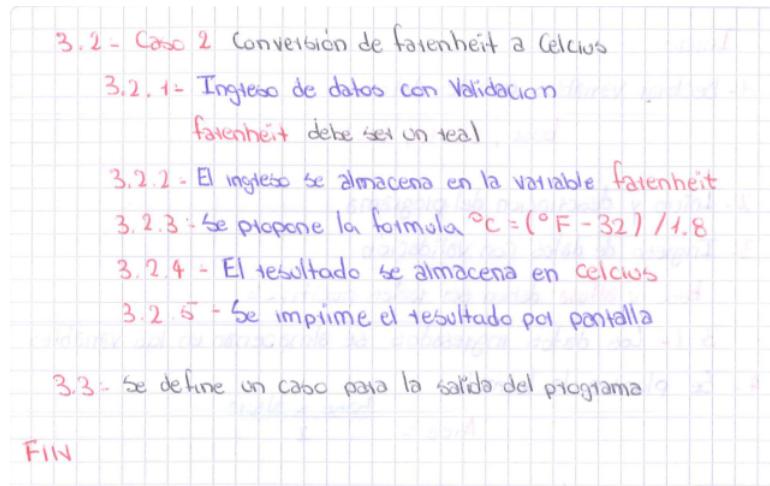
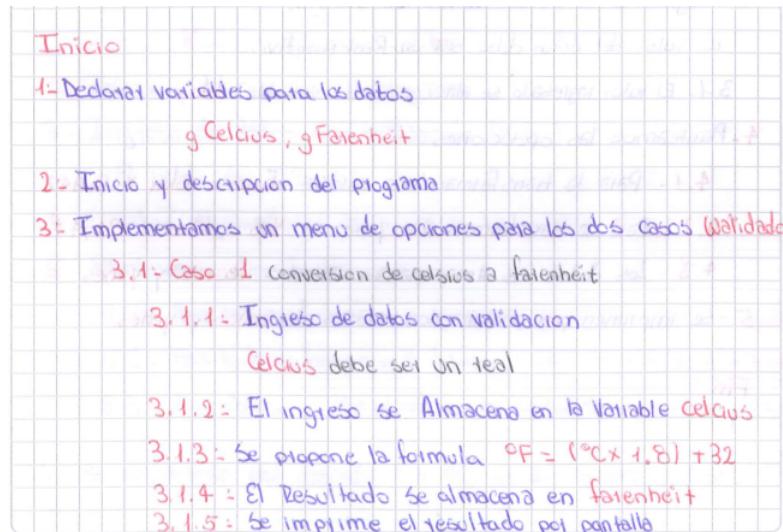
Formular un modelo: Hemos determinado tras el Analisis que resolver nuestro problema no implica complejidad alguna, si trabajamos con los ya mencionados factores de conversion el calculo entre unidades se simplificaria a una sencilla operacion de suma/resta y multiplicacion/division. Se desarollara entonces el Analisis de los factores de conversion que se emplearan en la resolucion del ejercicio.

Transformación de Temperatura ${}^{\circ}\text{C} \rightarrow {}^{\circ}\text{F}$ y ${}^{\circ}\text{F} \rightarrow {}^{\circ}\text{C}$									
Consideramos									
<table border="1"> <thead> <tr> <th>Celsius</th> <th>Fahrenheit</th> </tr> </thead> <tbody> <tr> <td>${}^{\circ}\text{C}$</td> <td>${}^{\circ}\text{F}$</td> </tr> <tr> <td>Conversion</td> <td>$({}^{\circ}\text{C} \times 1.8) + 32$</td> </tr> <tr> <td></td> <td>$({}^{\circ}\text{F} - 32) / 1.8$</td> </tr> </tbody> </table>		Celsius	Fahrenheit	${}^{\circ}\text{C}$	${}^{\circ}\text{F}$	Conversion	$({}^{\circ}\text{C} \times 1.8) + 32$		$({}^{\circ}\text{F} - 32) / 1.8$
Celsius	Fahrenheit								
${}^{\circ}\text{C}$	${}^{\circ}\text{F}$								
Conversion	$({}^{\circ}\text{C} \times 1.8) + 32$								
	$({}^{\circ}\text{F} - 32) / 1.8$								
	Las Fórmulas de Conversión entre Temperaturas								
Tenemos 2 casos									
1- Conversión de Celsius a Fahrenheit $\rightarrow {}^{\circ}\text{F} = ({}^{\circ}\text{C} \times 1.8) + 32$									
2- Conversión de Fahrenheit a Celsius $\rightarrow {}^{\circ}\text{C} = ({}^{\circ}\text{F} - 32) / 1.8$									
Para el caso 1 la entrada se hace en valores Celsius, y para el caso 2 la entrada se hace en valores Fahrenheit									

Fig. 41. Análisis para la conversión entre unidades de Temperatura

Una vez desarrollado el Análisis que supondría un camino para el funcionamiento de nuestro programa podríamos pensar en una vía por la cual haríamos que el análisis suceda de manera ordenada y correcta. Para esto nos planteamos escribir una serie de pasos describiendo todos y cada uno de los pasos que se requerirían para realizar la Transformación entre unidades de Temperatura utilizando Factores de Conversión, esta serie de instrucciones no es más que el Pseudocódigo de nuestro Análisis.

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

**Escribe el programa:**

En el Listing 1.9. Se encuentra el desarrollo del algoritmo cuyo objetivo es Transformar entre si las Unidades de Temperatura planteadas durante el Análisis.

```

1
2      """
3  Descripción:
4  Ejercicio 9.

```

```
5  Ingresar la temperatura en grados Celsius y convertirla en
   grados Fahrenheit y viceversa .
6  Autor:
7  Erick Sebastian Mora
8  """
9  def validarNumeros():
10     """
11         Es un procedimiento generico alternativo para realizar
12             validacion de numeros enteros
13             los datos ingresados se leen como string (str) y si
14                 son numeros se verifican
15                 y se transforman en tipo de datos entero (int), en
16                     este variante de funcion
17                     se maneja el uso de una excepcion para el ingreso de
18                         datos que no son numericos:
19
20             No Recibe parametros de entrada
21
22             Retorna:
23
24             Retorna un valor entero (numero) que es el dato
25                 ingresado transformado a (int)
26             """
27             while True:
28                 num = input()
29                 try:
30                     num = int(num)
31                     return num
32                 except ValueError:
33                     print("Solo se aceptan enteros")
34
35             def celsiusFarenheit():
36                 print("Ingrese la Temperatura en Celsius para "
37                     "Transformar a Fahrenheit:")
38                 celsius = validarNumeros()
39                 farenheit = (celsius * 1.8) + 32
40                 print(celsius , " C ", "equivale a ", farenheit , " F ")
41
42             def farenheitCelsius():
43                 print("Ingrese la Temperatura en Fahrenheit para "
44                     "Transformar a Celsius:")
45                 farenheit = validarNumeros()
46                 celsius = (farenheit - 32) / 1.8
47                 print(farenheit , " F ", "equivale a ", celsius , " C ")
48
49             def salir():
50                 print("Saliendo")
51                 exit()
52
53             def menuOpciones():
54                 """
55                     menu de opciones
56                     1. Conversión Celsius a Fahrenheit
57                     2. Conversión Fahrenheit a Celsius
58                     3. Salir
59                 """
60
61                 opcion = int(input("Seleccione una opción:"))
62
63                 if opcion == 1:
64                     celsiusFarenheit()
65                 elif opcion == 2:
66                     farenheitCelsius()
67                 elif opcion == 3:
68                     salir()
69
70             menuOpciones()
```

```
47     print("Vamos_a_Transformar_la_Temperatura_"
48         Ingresada")
49     print("1._Transformar_de_Celsius_a_Farenheit")
50     print("2._Transformar_de_Farenheit_a_Celsius")
51     print("3._Salir")
52     print("Ingrese_el_n_mero_de_la_opci_n_deseada:")
53     )
54     opcion = validarMenu()
55     return opcion
56
57 def validarMenu():
58     """
59     Es un procedimiento generico alternativo para realizar
60         validacion de numeros enteros
61     los datos ingresados se leen como string (str) y si
62         son numeros se verifican
63     y se transforman en tipo de datos entero (int), en
64         este variante de funcion
65     se maneja el uso de una excepcion para el ingreso de
66         datos que no son numericos:
67
68     No Recibe parametros de entrada
69
70     Retorna:
71
72     Retorna un valor entero (numero) que es el dato
73         ingresado transformado a (int)
74     """
75     while True:
76         num = input()
77         try:
78             num = int(num)
79             if 1<= num <=3:
80                 return num
81             else:
82                 print("Opcion_no_valida..Intentelo_de_"
83                     Nuevo")
84         except ValueError:
85             print("Solo_se_aceptan_Reales")
86
87 def seleccionOpcion(opcion):
88     if opcion == 1:
89         celsiusFarenheit()
90     elif opcion == 2:
91         farenheitCelsius()
92     elif opcion == 3:
93         salir()
```

```

89  if __name__ == '__main__':
90      nuevoMenu = menuOpciones()
91      seleccionOpcion(nuevoMenu)

```

Listing 1.9. Desarrollo del ejercicio 9

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema.

Celcius	Farenheit	
42	-	ENTRADA
42	$F = (C * 1.8) + 32$	1 ITERACION
42	107.6	RESULTADO

Table 49. Caso de Prueba 1

Celcius	Farenheit	
-	23	ENTRADA
$C = (F - 32) / 1.8$	23	1 ITERACION
-5	23	RESULTADO

Table 50. Caso de Prueba 2

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de escritorio manuales, el resultado debería suponer similitud con los datos obtenidos en las

Celcius	Farenheit	
-	52	ENTRADA
$C = (F-32)/1.8$	52	1 ITERACION
11.11	52	RESULTADO

Table 51. Caso de Prueba 3

pruebas de escritorio.

Para el Caso de Prueba 1. 52

Celcius	Farenheit	
42	-	ENTRADA
42	$F = (C*1.8)+32$	1 ITERACION
42	107.6	RESULTADO

Table 52. Caso de Prueba 1

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 42

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

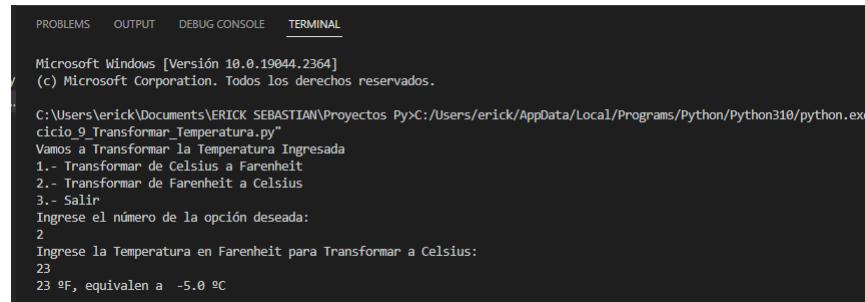
C:\Users\erick\Documents\ERTCK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_9_Transformar_Temperatura.py"
Vamos a Transformar la Temperatura Ingresada
1.- Transformar de Celsius a Farenheit
2.- Transformar de Farenheit a Celsius
3.- Salir
Ingrese el número de la opción deseada:
1
Ingrese la Temperatura en Celsius para Transformar a Farenheit:
42
42 °C, equivalen a 107.6000000000001 °F

```

Fig. 42. Ejecucion 1: Ejercicio 9

Para el Caso de Prueba 2. 53
Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 43

Celcius	Farenheit	
-	23	ENTRADA
$C = (F-32)/1.8$	23	1 ITERACION
-5	23	RESULTADO

Table 53. Caso de Prueba 2


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ejercicio_9_Transformar_Temperatura.py"
Vamos a Transformar la Temperatura Ingresada
1.- Transformar de Celcius a Farenheit
2.- Transformar de Farenheit a Celcius
3.- Salir
Ingrese el número de la opción deseada:
2
Ingrese la Temperatura en Farenheit para Transformar a Celcius:
23
23 °F, equivalen a -5.0 °C

```

Fig. 43. Ejecucion 2: Ejercicio 9

Para el Caso de Prueba 3. 54

Celcius	Farenheit	
-	52	ENTRADA
$C = (F-32)/1.8$	52	1 ITERACION
11.11	52	RESULTADO

Table 54. Caso de Prueba 3

Se evalua la variable de entrada y se obtienen los siguientes resultados. Véase 44

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ejercicio_9_Transformar_Temperatura.py"
Vamos a Transformar la Temperatura Ingresada
1.- Transformar de Celsius a Farenheit
2.- Transformar de Farenheit a Celsius
3.- Salir
Ingrese el número de la opción deseada:
2
Ingrese la Temperatura en Farenheit para Transformar a Celsius:
52
52 °F, equivalen a 11.11111111111111 °C

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 44. Ejecucion 3: Ejercicio 9

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, en la ejecución y compararlos con los datos obtenidos en nuestras pruebas manuales, si existe similitud entre estos datos entonces podemos dar por concluido nuestro problema. Particularmente en este caso hemos evidenciado que tanto los datos obtenidos en el programa como los de nuestras pruebas de escritorio coinciden. Por lo tanto damos por concluido el ejercicio.

2.10 Ejercicio 1: Calcular el Área de un Triangulo

Dados dos datos ingresadas por el usuario que corresponden a la base y altura de un Triangulo x, calcular el área del mismo.

Comprensión del problema: El triangulo es una figura geométrica de la clasificación de los polígonos que al igual que otras formas geométricas, cuentan con sus dimensiones. Particularmente, si nos referimos a los triángulos estos tienen las suyas bien definidas y las más importantes las conocemos como base y altura, dado que conocemos que el triángulo es una composición de 3 lados y 3 vértices y que de los 3 lados podemos destacar a la base como el más importante, dado que a partir de la base podemos calcular la altura de la forma geométrica trazando una línea perpendicular desde la base hasta el vértice opuesto.

Sin embargo, los lados no son los únicos datos que se pueden obtener dado un triángulo definido, a partir de su base y su altura se puede calcular dimensiones como su área. El área no es más que el valor del espacio contenido entre los 3 lados.

Formular un modelo: Para poder implementar un modelo algorítmico que calcule los datos de un triángulo haciendo uso del lenguaje Python, es necesario tener claro que es lo que tenemos y que es lo que buscamos. El problema dicta

que debemos conocer la base y altura del triángulo, el solo conocer estas dos dimensiones será información suficiente para realizar los cálculos pertinentes para el problema.

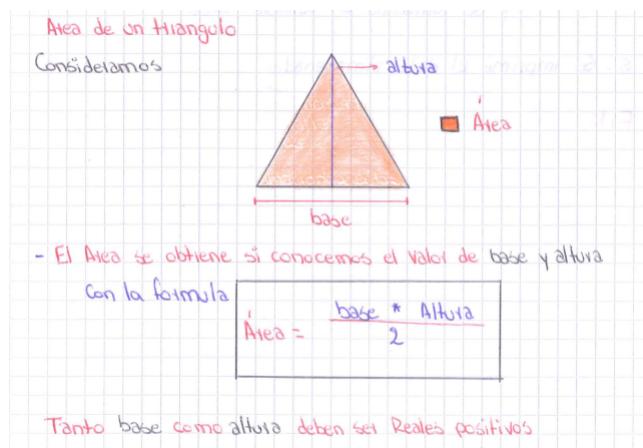
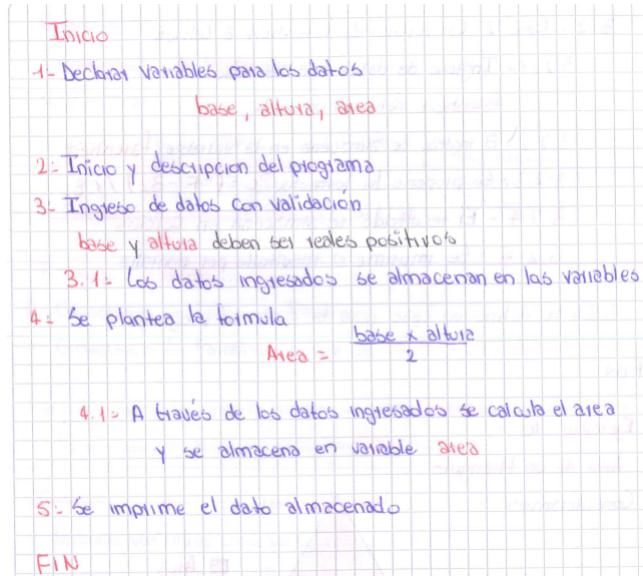


Fig. 45. Análisis de los Datos de un Triángulo

Desarrollar el modelo: A continuación se ha desarrollado un modelo escrito utilizando pseudocódigo, para describir el problema y su camino para llegar a la solución.

**Fig. 46.** Pseudocódigo del Proceso para obtener el Área de un Triángulo**Escribe el programa:**

En el Listing 1.10. Se encuentra el desarrollo del algoritmo cuyo objetivo es calcular el valor del Área de un triángulo ingresando sus valores (base y altura).

```

1
2      """
3  Descripción:
4  Ejercicio 10.
5  Ingresar la base y la altura de un triángulo y hallar su
     .area .
6  Autor:
7  Erick Sebastian Mora
8  """
9 def validarNumeros():
10     """
11     Es un procedimiento generico alternativo para realizar
     .validacion de numeros enteros
12     los datos ingresados se leen como string (str) y si
     .son numeros se verifican
13     y se transforman en tipo de datos entero (int), en
     .este variante de funcion
14     se maneja el uso de una excepcion para el ingreso de
     .datos que no son numericos:
15
16     No Recibe parametros de entrada
17

```

```

18     Retorna:
19     _____
20         Retorna un valor entero (numero) que es el dato
21             ingresado transformado a (int)
21         """
22     while True:
23         num = input()
24         try:
25             num = float(num)
26             if num >0:
27                 return num
28             else:
29                 print("Solo numeros mayores a 0")
30         except ValueError:
31             print("Solo se aceptan Reales")
32
33     def ingresarDatos():
34         print("Vamos a calcular el Area del Triangulo . . .
35             Necesitamos el valor de su Base y su Altura : . . ")
36         print("Ingrese el valor de la Base del Triangulo : . . ")
37         baseTr = validarNumeros()
38         print("Ingrese el valor de la Altura del Triangulo : . . ")
39         alturaTr = validarNumeros()
40         datos = [baseTr, alturaTr]
41         return datos
42
42     def calcularArea():
43         datos = ingresarDatos()
44         baseTr = datos[0]
45         alturaTr = datos[1]
46
47         areaTr = (baseTr*alturaTr)/2
48
49         print("El area del triangulo es : . . . ,round(areaTr ,2) )
50
51     if __name__ == '__main__':
52         calcularArea()

```

Listing 1.10. Desarrollo del ejercicio 10

Link del código en el repositorio de github: https://github.com/Erickxse/-8001-_Mora_Erick_Ejercicios_Python.git

Pruebe el programa: A continuación se pondrá a prueba el algoritmo generado en el lenguaje Python, para lo cual se ha considerado pertinente realizar pruebas de escritorio previo a la ejecución y evaluación del programa. Para lo cual se presentan a continuación 3 diferentes casos de pruebas de escritorio correspondientes al problema. Véase 55 , 56 , 57

Ejecución: Una vez desarrollado nuestro algoritmo en código Python podemos ponerlo a prueba tomando en cuenta los datos de nuestros 3 ensayos de es-

Base	Altura	Area	
3	5	-	ENTRADA
3	5	$A = (bxh)/2$	1 ITERACION
3	5	7.5	RESULTADO

Table 55. Caso de Prueba 1

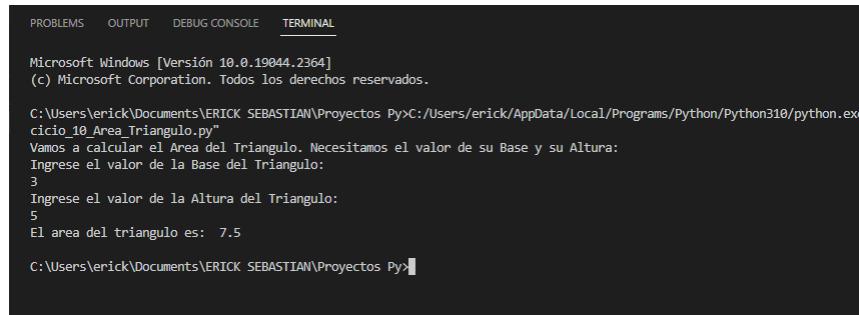
Base	Altura	Area	
6	3	-	ENTRADA
6	3	$A = (bxh)/2$	1 ITERACION
6	3	9	RESULTADO

Table 56. Caso de Prueba 2

citorio manuales, el resultado debería suponer similitud con los datos obtenidos en las pruebas de escritorio.

Para el Caso de Prueba 1. 55

Se evaluan los datos y se obtienen los siguientes resultados. Véase 47



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_10_Area_Triangulo.py"
Vamos a calcular el Área del Triangulo. Necesitamos el valor de su Base y su Altura:
Ingrese el valor de la Base del Triangulo:
3
Ingrese el valor de la Altura del Triangulo:
5
El area del triangulo es: 7.5
C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

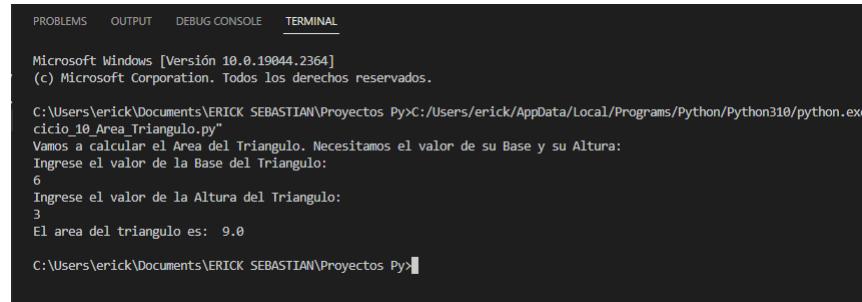
```

Fig. 47. Ejecucion 1: Ejercicio 10

Para el Caso de Prueba 2. 56 Se evaluan los datos y se obtienen los siguientes resultados. Véase 48

Base	Altura	Area	
4.6	3.2	-	ENTRADA
4.6	3.2	$A = (bxh)/2$	1 ITERACION
4.6	3.2	7.36	RESULTADO

Table 57. Caso de Prueba 3



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_10_Area_Triangulo.py"
Vamos a calcular el Área del Triángulo. Necesitamos el valor de su Base y su Altura:
Ingrese el valor de la Base del Triángulo:
6
Ingrese el valor de la Altura del Triángulo:
3
El área del triángulo es: 9.0

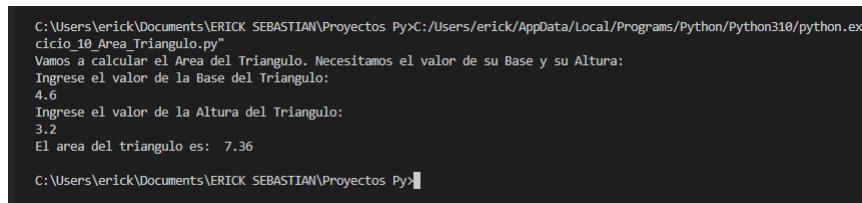
C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 48. Ejecucion 2: Ejercicio 10

Para el Caso de Prueba 3. 56

Se evaluan los datos y se obtienen los siguientes resultados. Véase 49



```

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>C:/Users/erick/AppData/Local/Programs/Python/Python310/python.exe
ciclo_10_Area_Triangulo.py"
Vamos a calcular el Área del Triángulo. Necesitamos el valor de su Base y su Altura:
Ingrese el valor de la Base del Triángulo:
4.6
Ingrese el valor de la Altura del Triángulo:
3.2
El área del triángulo es: 7.36

C:\Users\erick\Documents\ERICK SEBASTIAN\Proyectos Py>

```

Fig. 49. Ejecucion 3: Ejercicio 10

Evalúe la solución: Una vez ejecutado el programa hay que proceder a evaluar cada uno de los resultados obtenidos, los cuales deben proporcionar una similitud total con los obtenidos en las pruebas manuales realizadas en el Análisis. Si la evaluación es correcta el ejercicio cumple con su objetivo y por lo tanto se lo da por realizado.

3 Discusión

Durante el desarrollo del proyecto se evidencio que Python es una herramienta sumamente poderosa cuando se trata de resolver formulas y problemas relacionados con las matematicas basicas, y ademas de su potencia presenta una alta simpleza en su sintaxis lo que facilita el aprendizaje del Lenguaje para cualquier persona

En la resolución de este proyecto, pudimos aplicar algunos conocimientos de lógica proposicional y Análisis de problemas obtenidos en la materia de Modelos Discretos, para poder desarrollar la serie de ejercicios planteados en el lenguaje de programación Python.

Ciertos modelos aplicados en el presente trabajo, presentaron en su composición aplicaciones de fórmulas lógicas y matemáticas, algunas con más dificultad que otras, sin embargo todas las operaciones utilizadas por mínimas que sean le brindan sentido a nuestro algoritmo.

4 Conclusiones

En el desarrollo de este proyecto se logró crear un total de diez programas usando el lenguaje de programación Python, en el cual simulamos una gama de problemas y calculos matematicos, para el cual se adaptaron para usar el Análisis en su desarrollo, modelado e implementación.

Como se puede observar en el presente proyecto se ha aplicado conocimientos adquiridos en semestres anteriores, pero sobre todo podemos sintetizar que la lógica proposicional en conjunto con el lenguaje de programación python permiten un mayor entendimiento de la aplicación de la materia en el campo del desarrollo de software.

References

1. Néstor Jiménez, Carlos Pizarro, and Jesús Cortes. Círculo y circunferencia. 1977.
2. Andrés Marzal Varó, Isabel Gracia Luengo, et al. *Introducción a la programación con Python*. Universitat Jaume I, 2009.