

# Reporte de Laboratorio Nro. 2

Dayanna Jamilex Reyes Navarrete<sup>L00081418</sup>

Universidad de las Fuerzas Armadas ESPE  
djreyes4@espe.edu.ec

Tema: Entorno de Desarrollo y Versionamiento

## Resumen

En el presente laboratorio se llevará a cabo el desarrollo de un ejercicio matemático, el cual será calcular el área de un triángulo equilátero. Para realizar dicho calculo se hará su respectivo proceso haciendo uso de su debida fórmula. Este ejercicio se realizará haciendo uso del lenguaje de programación Python. El cual se realizará mediante funciones y procedimientos para mejor comprensión del mismo. Dicho programa será entendible para el usuario y de fácil manejo, esto se logrará haciendo uso de las buenas prácticas de programación, usando las debidas variables con su respectivo tipo de dato las cuales serán claras y precisas. Adicional el programa sera versionado creando un nuevo repositorio el github.

## 1. Introducción

Python es un lenguaje de programación de muy alto nivel, con una sintaxis muy clara y un fuerte compromiso con la legibilidad del código. Definitivamente es un lenguaje de programación muy versátil, fuertemente tipado, imperativo y orientado a objetos. Aunque también incluye la capacidad de convertirlo en un lenguaje de paradigma funcional. [2] En la actualidad, Python es considerado uno de los lenguajes que más facilidades ofrece para enseñar programación orientada a objetos.

Para la programación orientada a objetos (POO) necesitamos aprender diferentes formas de pensar sobre como poder programar en términos de objetos, propiedades y métodos. La POO no es difícil, pero es una manera especial de pensar, y puede ser subjetivo del programador, por lo que diferentes programadores pueden hacerlo de manera diferente. Hay muchas formas de programar, pero no todas son correctas. Lo difícil es programar bien. La programación adecuada es importante para aprovechar al máximo la POO de esta manera [1].

El objetivo del presente laboratorio es aplicar los conocimientos adquiridos referente a funciones y procedimientos en el lenguaje de programación Python. Para llevar a cabo el objetivo se realizará un ejercicio matemático, el cual será calcular el área de un triangulo equilátero. En donde se hará uso de funciones y procedimientos, y de esta manera se dará solución al problema planteado. Adicional se mostrara los posibles errores que se pueden cometer durante este programa, y de esta manera se mostara la solucione de dichos errores.

## 2. Método

A continuación se presenta el ejercicio planteado, el cual será resuelto aplicado lo aprendido en clase, que son las funciones y procedimientos. Para resolver dicho ejercicio se realizara una serie de pasos donde se dara a conocer como se llego a la solucion del mismo.

### 2.1. Comprender el problema

Para la resolucion del problema primero se debe reconocer que tipo de triangulo vamos a utilizar, una vez identificado el triangulo se debe tener entender la formula para calcular el area del mismo. Para esto se realizará un programa que permita al usuario ingresar la base de un triangulo la cual sera un numero cualquiera ya sea entero o decimal, de la misma manera se ingresara la altura del triangulo. Este se realiza utilizando una funcion en Python, ya que las funciones nos ayuda hacer en partes un problema.

### 2.2. Formular un modelo

Lo primero que vamos a hacer es a analizar el problema que tenemos que resolver, en este caso como bien podemos observar en la figura 1 tenemos un triangulo equilatero y como bien sabemos este tiene todos sus lados iguales. Para calcular el area lo que nesecitamos saber es su base y su altura, ya que la formula para calcular el area es,  $\text{base} \times \text{altura}$  y esto dividido para 2, de esta manera obtendremos el resultado de la altura.

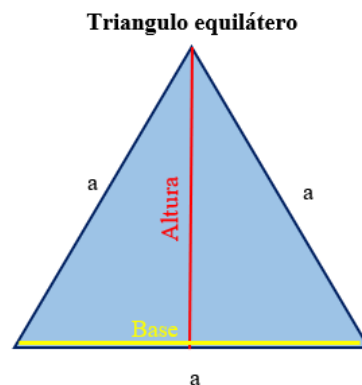


Figura 1: Triangulo equilatero.

### 2.3. Desarrollar un algoritmo

Una vez que se formulo el modelo se procede a plantear la idea en un algoritmo, este nos ayuda a entender de mejor manera el problema, ya que los algoritmos son una secuencia pasos finitos, ordenados, que nos lleva a resolver un problema. Como se puede observar en la figura 2 se ha realizado el respectivo pseudocodigo donde se encuentra la secuencia de pasos ordenados, la cual contiene un inicio, un proceso, y un fin, el cual cumple con la estructura del algoritmo, de esta manera obtendremos el resultado esperado.

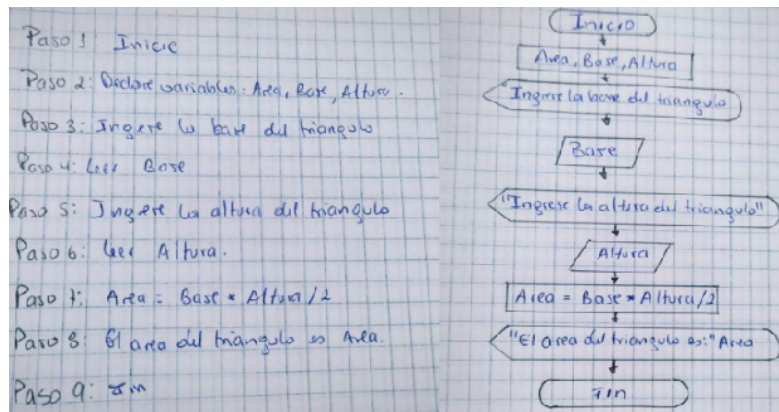


Figura 2: Pseudocódigo y algoritmo.

## 2.4. Escribe el programa

Una vez hecho el algoritmo se procede a transformarlo en instrucciones que la computadora logre entender. Para llegar a este paso se realizó un análisis y un algoritmo el cual nos facilitará comprender mejor el programa. Como se puede observar en la figura 3 se hizo uso de una función (AreaTriangulo) y de sus respectivas variables las cuales son, base, altura que son de tipo float, debido a que los lados o altura puede tener cualquier valor, ya sea entero y decimal, se aplicó también la debida fórmula para calcular el área.

Aplicando también las buenas prácticas de programación, como comentar el código para mayor facilidad de entendimiento y teniendo buena indentación, ya que la indentación del código es uno de los errores más comunes al programar.

```

1  #Calcular el area de un triangulo equilatero
2  #Se declara las variables Base y Altura
3  base=float(input("Ingrese la base del triangulo: ")) #Se pide ingresar los datos pedidos en el programa
4  altura=float(input("Ingrese la altura del triangulo: "))
5  #Funcion para calcular el area del triangulo
6  def Areatriangulo():
7      #Formula para calcular el area
8      Area=base*altura/2
9      print("El area del triangulo equilatero es: ")
10     #Se retorna el area
11     return Area
12     #Se llama a la funcion
13 Areatriangulo()
  
```

Figura 3: Código fuente.

## 2.5. Pruebe el programa

Tras haber realizado anteriormente el código se procede a realizar la prueba del mismo, el cual se realiza mediante python usando el IDE visual studio code, se realizarán 5 casos de prueba:

**Caso de prueba 1** Tenemos la primera prueba en el cual se ingreso numero enteros en el que se ingreso como base 12 y altutura 15 y nos da como resultado 90.0 el cual esta correcto, tal como se muestra en la figura 4.

```
PS C:\Users\djrn4\Desktop\Pyton> & C:/Users/djrn4/AppData/Local/Programs/Python/Python310/python.exe es.py
Ingrese la base del triangulo: 12
Ingrese la altura del triangulo: 15
El area del triangulo equilatero es: 90.0
PS C:\Users\djrn4\Desktop\Pyton>
```

Figura 4: Primer caso de prueba.

**Caso de prueba 2** Tenemos la segunda prueba en el cual se ingreso numero decimales en el que se ingreso como base 5.8 y altutura 9.4 y nos da como resultado 27.26 el cual esta correcto, tal como se muestra en la figura 5.

```
PS C:\Users\djrn4\Desktop\Pyton> & C:/Users/djrn4/AppData/Local/Programs/Python/Python310/python.exe es.py
Ingrese la base del triangulo: 5.8
Ingrese la altura del triangulo: 9.4
El area del triangulo equilatero es: 27.26
PS C:\Users\djrn4\Desktop\Pyton>
```

Figura 5: Segundo caso de prueba.

**Caso de prueba 3** Tenemos la tercera prueba en el cual se ingreso numero enteros en el que se ingreso como base 85 y altutura 150 y nos da como resultado 6375.0 el cual esta correcto, tal como se muestra en la figura 6.

```
PS C:\Users\djrn4\Desktop\Pyton> & C:/Users/djrn4/AppData/Local/Programs/Python/Python310/python.exe es.py
Ingrese la base del triangulo: 85
Ingrese la altura del triangulo: 150
El area del triangulo equilatero es: 6375.0
PS C:\Users\djrn4\Desktop\Pyton>
```

Figura 6: Tercer caso de prueba.

**Caso de prueba 4** Tenemos la tercera prueba en el cual se ingreso numero enteros en el que se ingreso como base 0.04 y altutura 0.09 y nos da como resultado 0.0018 el cual esta correcto, tal como se muestra en la figura 7.

```

PS C:\Users\djrn4\Desktop\Pyton> & C:/Users/djrn4/AppData/Local/Programs/Python/Python310/python.exe es.py
Ingrese la base del triangulo: 0.04
Ingrese la altura del triangulo: 0.09
El area del triangulo equilatero es: 0.0018
PS C:\Users\djrn4\Desktop\Pyton>

```

Figura 7: Cuarto caso de prueba.

**Caso de prueba 5** Tenemos la tercera prueba en el cual se ingreso numero enteros en el que se ingreso como base 20 y altutura 29 y nos da como resultado 290.0 el cual esta correcto, tal como se muestra en la figura 8.

```

PS C:\Users\djrn4\Desktop\Pyton> & C:/Users/djrn4/AppData/Local/Programs/Python/Python310/python.exe es.py
Ingrese la base del triangulo: 20
Ingrese la altura del triangulo: 29
El area del triangulo equilatero es: 290.0
PS C:\Users\djrn4\Desktop\Pyton>

```

Figura 8: Quinto caso de prueba.

## 2.6. Evalúe la solución

Por ultimo se analizara la verificacion dada debido a las pruebas realizadas, po lo que se hace un breve analisis del problema original, con esto se puede comprobar el que el programa funciono correctamente, ya que nos dio los resultados esperados. Como se puede observar en la figura 9 Se realizó el análisis del problema original.

```

Get Started  AreaTrianguloReyes.py
AreaTrianguloReyes.py > ...
1  #Calcular el area de un triangulo equilatero
2  #Se declara las variables Base y Altura
3  base=float(input("Ingrese la base del triangulo: ")) #Se pide ingresar los datos pedidos en el programa
4  altura=float(input("Ingrese la altura del triangulo: "))
5  #Funcion para calcular el area del triangulo
6  def Areatriangulo():
7      #Formula para calcular el area
8      Area=base*altura/2
9      print("El area del triangulo equilatero es: ")
10     #Se retorna el area
11     return Area
12     #Se llama a la funcion
13     Areatriangulo()

```

Figura 9: Evalua la solución.

## 3. Versionamiento

Procedemos a crear un nuevo repositorio sin usar README, licencia o gitignore, para evitar posibles errores, podemos obserbar en la figura 10 la creacion del repositorio.


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository els [Import a repository](#).

---

Owner \*

Repository name \*

 DayannaReyes01 ▾


 / 

Lab2\_Reyes\_Dayanna ✓

Great repository names are short and memorable. Need inspiration? How about [silver-octo-giggle?](#)

Description (optional)

---

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

Figura 10: Creación del repositorio.

**Cambio del directorio de trabajo** Una vez creado el repositorio, abrimos git bash y inicializamos el directorio local como un repositorio de Git como se muestra en la figura 11. Aconti-

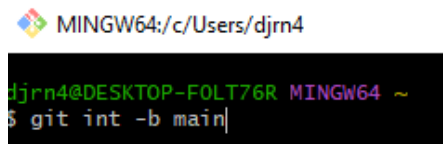


Figura 11: Inicializar Directorio local como repositorio de Git.

nuacion ingresamos los respectivos comando para realizar el cambio de directorio, los comando a ingresa estan en la figura 12

### ...or create a new repository on the command line

```
echo "# -NRC_6181-DayannaReyes_Lab2U1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/DayannaReyes01/-NRC_6181-DayannaReyes_Lab2U1.git
git push -u origin main
```

Figura 12: Comando Git.

Se debe ingresar los comando de manera correcta, para no tener problemas durante el proceso. De esta manera obtendremos el laborito en el repositorio de GitHub exitosamente. Adicionalmente se adjunta link de GitHub:

## 4. Results and Analysis

Los resultados obtenidos en este laboratorio han sido exitosos. porque se logró encontrar una solución al problema planteado, se obtuvo una solución precisa que me ayudó a lograr los resultados esperados. Para la resolución de dicho problema se logró haciendo uso de una serie de pasos. En el problema planteado se hizo uso de una función, la cual ayudó a resolver el problema. También se hizo uso de dos variables las cuales fueron Base y altura, estas variables se multiplicaron y se dividió para 2 y así logramos obtener el área del triángulo. De tal dicho problema fue versionado y subido al repositorio de GitHub exitosamente.

## 5. Discusión

Usando el lenguaje de programación python se llevó a cabo el desarrollo de un subprograma en la cual dentro de las funciones se encuentran dos connotaciones semánticas de alta relevancia, de la misma manera haciendo uso correcto de las variables. Esto nos ayuda a la resolución de problemas sin errores. A continuación se mostrará una serie de preguntas respecto a lo ya realizado anteriormente.

## 6. Sección de preguntas

### 6.1. ¿Enumere los 6 pasos para la resolución de problemas?

1. Comprender el problema: Se realiza un análisis del problema, 2. Formular un modelo: una vez que se analizó el problema se crean fórmulas para posibles resultados, 3. desarrollar un algoritmo: se crea una secuencia de pasos que contenga entrada proceso y fin, 4. Escribir el programa: se escribe el programa en código, 5. probar el programa: se ejecuta el programa para comprobar que no haya errores, 6. Evalúe la solución: se verifica que los resultados obtenidos sean los esperados desde un inicio.

### 6.2. ¿Defina el concepto de resolución de problemas?

La resolución de problemas asistida por computadora es el proceso que conduce al desarrollo de un programa que resuelve un problema, a partir de la explicación del problema, generalmente en lenguaje natural, expresado en términos específicos del dominio del problema. Es una serie de procesos para analizar la información relacionada con una situación dada, para generar las opciones de respuesta adecuadas.

### 6.3. ¿Qué es un algoritmo?

Es una secuencia de pasos finitos ordenados que nos llevan a la solución de problemas o a la toma de decisiones. Los algoritmos se utilizan a menudo en matemáticas y lógica, son la base para la creación de manuales de usuario, manuales de instrucciones, entre otros.

### 6.4. ¿Cómo puede definir usted el proceso de compilación?

Esto significa que el código que escribimos se traduce en código que la máquina puede entender, como una computadora entiende un código binario que consta de 0 y 1. El compilador toma todo el código y lo traduce al lenguaje máquina, si este tiene errores se detiene, se debe corregir y se vuelve a traducir el código.

### 6.5. ¿Explique el proceso de ejecución del programa?

Es el proceso de decirle a la computadora para evaluar las instrucciones compiladas. Cuando se ejecute el programa, si todo está bien debería ver la salida correcta. Cuando en la ejecución hay errores se debe corregir el código volver a ejecutar y comprobar.

### 6.6. ¿Comente sobre los errores que se pueden hallar al ejecutar un programa?

Errores de sintaxis, también se debe tomar mucho la indentación del código, otro de los errores más comunes es la división para 0, También el uso inadecuado del programa por parte del usuario, datos incorrectos o están en un formato diferente al esperado, entre otros.

### 6.7. ¿Por qué es importante el proceso de versionamiento?

El control de versiones toma una instantánea del archivo durante todo el proceso de creación. Estas instantáneas muestran cuándo se tomaron y qué cambios ocurrieron entre cada instantánea, lo que le permite restaurar versiones anteriores de los archivos.

## 7. Conclusión

Con lo mencionado ya anteriormente con el uso de funciones en el lenguaje de programación python. El cual nos ayudó a entender de manera más fácil el problema a resolver. En el laboratorio se llevó a cabo gracias a los conocimientos adquiridos en clase, también se logró obtener el resultado esperado siguiendo la serie de pasos que se encontraban en la guía. De esta manera se obtuvo un programa exitoso.

También se realizó exitosamente el versionamiento del laboratorio, esto se logró siguiendo los respectivos pasos y aplicando lo explicado en clase anteriormente. Adicionalmente se realizó una serie de preguntas las cuales fueron respondidas con lo aprendido en este laboratorio, lo cual se hizo fácil de responder gracias al desarrollo del mismo.

## Referencias

- [1] Miguel A Alvarez. Qué es la programación orientada a objetos. *Desarrolloweb*. <https://desarrolloweb.com/articulos/499.php> (acceso: 12 de junio de 2019), 2001.
- [2] David Masip Rodó. El lenguaje python.