Programación de software.
Centro de Comercio y Turismo.
Regional Quindío. 2021
Instructor: Germán Alberto Angarita Henao

UNA MANERA DE ENFRENTAR PROBLEMAS EN PROGRAMACIÓN

A continuación, se sugerirá una manera ordenada de resolver problemas a través de la programación. También, el uso del documento **HERRAMIENTAS PARA LA LÓGICA CON PYTHON.**

Para resolver problemas en programación seguiremos los siguientes pasos:

1. Analizar el problema determinando qué respuesta debe dar nuestro algoritmo a construir y cuál es el tipo de ésta.
2. (ENTRADA DE DATOS)Leer el problema que debemos resolver y extraer lo siguiente:
    - Datos que requiere el algoritmo para trabajar.
    - De los datos que requiere nuestro algoritmo, ¿cuáles nos los proporciona el enunciado del problema ?
    - De los datos que requiere nuestro algoritmo, ¿cuáles no los proporciona el enunciado del problema y debemos obtenerlos de otras fuentes?
3. (PROCESO)¿Cuáles son las acciones que debe realizar nuestro algoritmo una vez le hemos proporcionado los datos que necesita para trabajar? Estas acciones se deben describir y ser enumeradas.
4. (SALIDA DE DATOS)Construir la respuesta de nuestro algoritmo

Para expresar los pasos anteriores escribiremos pseudocódigo, ya sea siguiendo un estándar o no, algo muy personal.

Veamos un ejemplo:

Problema: **Realizar un algoritmo para determinar la distancia de dos puntos en el espacio bidimensional.**

Contacto: gangaritah@gmail.com

Analizando el problema, podemos determinar que la respuesta de nuestro algoritmo debe ser numérica ya que se nos pide "**determinar la distancia":**

tipo salida: numérica

(ENTRADA DE DATOS)

Los datos que requiere nuestro algoritmo para trabajar son:
Como se habla de distancia entre dos puntos en el espacio bidimensional, o sea, el plano cartesiano, necesitaremos las coordenadas de dos puntos en el espacio, las cuales no las proporciona el problema, por lo cual, el usuario deberá ingresarlas por teclado.

coordenada_x1: lo ingresará el usuario
coordenada_y1: lo ingresará el usuario
coordenada_x2: lo ingresará el usuario
coordenada_y2: lo ingresará el usuario

(PROCESO)

¿QUÉ SE DEBE HACER?

1. **Pedir por teclado las coordenadas de los puntos**
2. **Calcular la distancia entre los puntos obtenidos**

¿CÓMO SE HACE LO QUE SE DEBE HACER?

Programación de software.
Centro de Comercio y Turismo.
Regional Quindío. 2021
Instructor: Germán Alberto Angarita Henao

Pedir por teclado las coordenadas de los puntos - se hace según caja de herramientas Python con el método *input*. Los datos ingresados deben ser convertidos con *float*, ya que operaremos con ellos.

Determinar la distancia entre los puntos obtenidos - se hace con la fórmula de distancia entre dos puntos $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ **Acá, notamos que la fórmula hace uso de raíz cuadrada y elevación al cuadrado, por lo tanto, necesitamos averiguar cómo se hacen estas dos operaciones en Python. Haciendo la investigación sabemos que:**
Para usar raíz cuadrada en Python podemos importar el módulo *math* y usar la función *sqrt*, la cual recibe un número y devuelve su raíz cuadrada:

```
import math
print(math.sqrt(4))#imprime 2, raiz cuadrada de 4
```

Para elevar un número al cuadrado en Python usar el operador de potenciación **, el cual toma un número y lo eleva a la potencia indicada:

```
print(2**3)#imprime 8, que es 2 elevado a la 3
print(2**2)#imprime 4, que es 2 elevado a la 2
```

(SALIDA DE DATOS)

¿QUÉ SE DEBE HACER?

Contacto: gangaritah@gmail.com

1. **Imprimir un mensaje al usuario mostrando la distancia entre los puntos tecleados por éste**

¿CÓMO SE HACE LO QUE SE DEBE HACER?

Imprimir un mensaje al usuario mostrando la distancia entre los puntos tecleados por éste - se hace con el método *print* según caja de herramientas Python

PASANDO NUESTRO PSEUDOCÓDIGO O CÓDIGO PYTHON

Ahora, debemos transcribir y adaptar las acciones expresadas en el análisis del problema a código Python:

(ENTRADA DE DATOS) Según lo planeado, las coordenadas de los puntos deben ser ingresadas por teclado por el usuario usando *input*. No podemos olvidar convertir estos datos a flotantes para poder operar con ellos.

```
coordenada_x1 = float(input("ingrese coordenada en x del primer punto"))
coordenada_y1 = float(input("ingrese coordenada en y del primer punto"))
coordenada_x2 = float(input("ingrese coordenada en x del segundo punto"))
coordenada_y2 = float(input("ingrese coordenada en y del segundo punto"))
```

(PROCESO) Según lo planeado, debemos aplicar la fórmula de distancia usando el método *sqrt* para trabajar con raíces cuadradas y usar el operador ** para elevar al cuadrado en este caso. El resultado que obtenemos al aplicar la fórmula de distancia lo guardamos en la variable *distancia*.


```
distancia = math.sqrt(
(coordenada_x2 - coordenada_x1)**2 + (coordenada_y2 - coordenada_y1)**2
    )
```

(SALIDA DE DATOS) Según lo planeado, se debe usar *print* para mostrar al usuario un mensaje indicando la distancia buscada. Recordar que el mensaje al usuario debe tener información del dato que se le muestra, de esta manera el usuario no se confundirá preguntándose qué significa el dato que nuestro programa le está mostrando.

```
print("La distancia entre los puntos es: ", distancia)
```

Por último, nuestro programa completo quedaría así(nótese que importamos el módulo math):

```
import math

coordenada_x1 = float(input("ingrese coordenada en x del primer punto"))
coordenada_y1 = float(input("ingrese coordenada en y del primer punto"))
coordenada_x2 = float(input("ingrese coordenada en x del segundo punto"))
coordenada_y2 = float(input("ingrese coordenada en y del segundo punto"))

distancia = math.sqrt(
(coordenada_x2 - coordenada_x1)**2 + (coordenada_y2 - coordenada_y1)**2
    )

print("La distancia entre los puntos es: ", distancia)
```

Ahora, aplicando los mismos pasos, resuelva en Python:

1. Realice un programa que calcule el índice de masa corporal de una persona.
2. Realice un programa que calcule el área de un trapecio isósceles.
3. Realice un programa que calcule la distancia entre dos puntos en el espacio tridimensional.