

A decorative graphic on the left side of the slide, consisting of a network of white lines and small circles on a blue gradient background, resembling a circuit board or a stylized tree structure.

# CALCULADORA TOPOGRÁFICA

# FUNÇÃO 1

- Calcula as coordenadas dos pontos desejados;
- Caso o usuário não possua as variações delta X e Y, calcula por outros dados.

```

def calculadora coordenadas():#função que calcula coordenadas
    print("Digite os valores de X e Y do ponto A:")#pede os valores ao usuário
    xa=float(input("X:"))#valor de x da coordenada inicial
    ya=float(input("Y:"))#valor de y da coordenada inicial
    r=input("Você possui as variações DeltaX e DeltaY?(S para sim, N para não)")#dá ao usuário a opção de
    if r == 'S':#validação da escolha do usuário
        DeltaX=float(input("DeltaX (em metros): "))#variação de metragem entre os pontos na coordenada x
        DeltaY=float(input("DeltaY (em metros): "))#variação de metragem entre os pontos na coordenada Y
        xb=DeltaX+xa#calcula a coordenada desejada em x
        yb=DeltaY+ya#calcula a coordenada desejada em y
        print("Xb= {0:.3f} ; Yb= {1:.3f}".format(xb,yb))#exibe ao usuário
    elif r == 'N':#validação da possível escolha do usuário
        g=int(input("Graus: "))#solicita os graus; int porque não admite grau decimal
        m=int(input("Minutos: "))#solicita os minutos; int porque não existe minuto decimal
        s=int(input("Segundos: "))#solicita os segundos, int porque não admite segundo decimal
        g1=(g+(m/60)+(s/3600))#conversor de grau decimal
        Dh=float(input("Distância Horizontal (em metros): "))#solicita a distância inclinada que será usada
        DeltaX=(math.sin(math.radians(g1)))*Dh#calcula DeltaX
        DeltaY=(math.cos(math.radians(g1)))*Dh#calcula DeltaY
        print("DeltaX={0:.3f} e DeltaY={1:.3f}".format(DeltaX,DeltaY))#imprime as variações ao usuário
        xb=DeltaX+xa#calcula a coordenada desejada em x
        yb=DeltaY+ya#calcula a coordenada desejada em y
        print("Xb= {0:.3f} ; Yb= {1:.3f}".format(xb,yb))#exibe ao usuário
    else :#rejeita respostas fora do permitido
        print("Erro - resposta deve ser apenas S ou N")
def calculadora de area(): #função que calcula área

```

PEP 8: at least two spaces before inline comment

PEP 8: inline comment should start with '#'

## FUNÇÃO 2

- É requisitado do usuário uma lista de coordenadas X e Y;
- A função while é utilizada para parar a lista de dados.

```

def calculadora_de_area(): #função que calcula área
    x=[]#lista vazia para os valores de x
    y=[]#lista vazia para os valores de y
    print("Insira uma lista de metragens de coordenadas X, ou digite x para finalizar a inserção.")#pede ao usuário os valores de X
    linha=input(">")#primeira solicitação de valores
    while linha != 'x':#laço para validar a continuação ou não da lista
        x.append(float(linha))#insere o que for digitado na lista x
        linha=(input(">"))#reinicia o processo
    x.append(float(x[0]))#repete o primeiro valor digitado, necessário para conta
    print("Insira uma lista de metragens de coordenadas Y, ou digite x para finalizar a inserção.")#pede aos usuários os valores de y
    linha=input(">")#primeira solicitação de valores
    while linha != 'x':#laço para validar a continuação ou não da lista
        y.append(float(linha))#insere o que for digitado na lista y
        linha=(input(">"))#reinicia o processo
    y.append(float(y[0]))#repete o primeiro valor digitado, necessário para conta
    i=0#contador, faz a lista "andar"
    somatoriox=0#armazena os valores multiplicados de x
    somatorioy=0#armazena os valores multiplicados de y
    while (i+1) < (len(x)):#calcula para todos os valores existentes na lista
        m=y[i]*x[(i+1)]#multiplicador em cruz
        somatoriox=somatoriox+m#soma
        i=i+1#movimenta o contador
    print("Somatório em x: {0:.3f}".format(somatoriox))#exibe o resultado
    i=0#zera o contador para que possa ser reaproveitado
    while (i+1) < (len(y)):#calcula para todos os valores existentes na lista
        m=x[i]*y[(i+1)]#multiplicador em cruz
        somatorioy=somatorioy+m#armazena
        i=i+1#anda com a lista
    print("Somatório em y: {0:.3f}".format(somatorioy))#resultado parcial
    a=(somatoriox-somatorioy)*0.5#finaliza a conta de área com a fórmula
    print("Área (em m^2): {0:.3f}".format(a))#exibe o valor em metros quadrados
    print("Área (em Hectares): {0:.3f}".format(a/10000))# e em hectares

def calculadora_de_nivelamento():#função que calcula os nivelamentos
    print("NIVELAMENTO - Insira os valores em metros:")
    calculadora_de_nivelamento()

```

## FUNÇÃO 3

- O cálculo de nivelamento é feito a partir de uma cadeia de condicionais que imprime na tela o valor de desnível requisitando do usuário os fios médios de ré e vante.
- O usuário interpretará o sinal do valor de acordo com a direção do desnível

