

# [Tarefa prática] Folha da árvore

Pontos escolhidos:

$x_0=1.6$      $y_0=0.5$

$x_1=3$      $y_1=1.6$

$x_2=5$      $y_2=2.3$

$x_3=6.8$      $y_3=2.1$

$x_4=8.3$      $y_4=1.4$

```
>_ Console × Shell +
Run Ask AI 36s on 21:42:37, 05/28 ✓

Digite a quantidade de pontos: 5
Digite o valor de x0: 1.6
Digite o valor de y0: 0.5
Digite o valor de x1: 3
Digite o valor de y1: 1.6
Digite o valor de x2: 5
Digite o valor de y2: 2.3
Digite o valor de x3: 6.8
Digite o valor de y3: 2.1
Digite o valor de x4: 8.3
Digite o valor de y4: 1.4
Escolha o método (para newton digite newton): newton

Quadro das Diferenças Divididas:
[0.5, 0.7857142857142858, -0.12815126050420175, 0.00130889062777
61026, 0.0001876598470415365]
[1.6, 0.34999999999999997, -0.12134502923976602, 0.0025662116029
543974, 0.0]
[2.3, -0.11111111111111098, -0.10774410774410771, 0.0, 0.0]
[2.1, -0.46666666666666665, 0.0, 0.0, 0.0]
[1.4, 0.0, 0.0, 0.0, 0.0]

Digite o ponto para calcular o polinômio: 6
O valor do polinômio em 6.0 é: 2.2808418867892803
```

O ponto escolhido foi o  $x = 6$  e o retorno foi igual a 2.2808418867892803

A seguir a imagem do ponto 6 sendo medido. O resultado tem uma precisão muito próxima ou até igual a da realidade.



A seguir o código utilizado para resolver o problema:

```
main.py × +
main.py > f calcular_diferencas_divididas > ... Format

1 def calcular_diferencas_divididas(x, y):
2     n = len(x)
3     dd = [[0.0] * n for _ in range(n)]
4     for i in range(n):
5         dd[i][0] = y[i]
6
7     for j in range(1, n):
8         for i in range(n - j):
9             dd[i][j] = (dd[i + 1][j - 1] - dd[i][j - 1]) / (x[i + j]
- x[i])
10
11     return dd
12
13 def calcular_polinomio_interpolador(x, y):
14     dd = calcular_diferencas_divididas(x, y)
15     n = len(x)
16
17     def polinomio_interpolador(ponto):
18         resultado = dd[0][0]
19         termo = 1.0
20
21         for i in range(1, n):
22             termo *= ponto - x[i - 1]
23             resultado += dd[0][i] * termo
24
25         return resultado
26
27     return polinomio_interpolador

Ln 1, Col 1 • Spaces: 4 History
```

```
main.py x +
main.py > f calcular_diferencas_divididas > ... Format

29 def main():
30     n = int(input("Digite a quantidade de pontos: "))
31
32     x = []
33     y = []
34
35     for i in range(n):
36         x_input = input(f"Digite o valor de x{i}: ")
37         x.append(float(x_input))
38         y_input = input(f"Digite o valor de y{i}: ")
39         y.append(float(y_input))
40
41     metodo = input("Escolha o método (para newton digite newton): ").lower()
42
43     if metodo == "newton":
44         polinomio = calcular_polinomio_interpolador(x, y)
45
46         print("\nQuadro das Diferenças Divididas:")
47         dd = calcular_diferencas_divididas(x, y)
48         for linha in dd:
49             print(linha)
50
51         ponto = float(input("\nDigite o ponto para calcular o polinômio: "))
52         resultado = polinomio(ponto)
53         print(f"\nO valor do polinômio em {ponto} é: {resultado}")
54
```

```
main.py x +
main.py > f calcular_diferencas_divididas > ... Form

55     else:
56         print("Método não reconhecido.")
57
58 if __name__ == "__main__":
59     main()
60
```