

Actividad 2 – II Unidad: Interpolación de Newton

1- EJERCICIO

```
1ejercicio1.py > ...
1 import numpy as np
2 def divided_differences(xs, ys):
3     n = len(xs)
4     table = np.zeros((n, n))
5     table[:,0] = ys
6     for j in range(1, n):
7         for i in range(n - j):
8             table[i,j] = (table[i+1,j-1] - table[i,j-1]) / (xs[i+j] - xs[i])
9         coef = table[0, :]
10    return coef, table
11 def newton_eval(coef, xs, x):
12    n = len(coef)
13    result = coef[n-1]
14    for k in range(n-2, -1, -1):
15        result = result * (x - xs[k]) + coef[k]
16    return result
17 xs = np.array([1.0, 2.0, 3.0, 4.0])
18 ys = np.array([1.0, 4.0, 9.0, 16.0])
19 coef, table = divided_differences(xs, ys)
20 print('Coeficientes (diferencias divididas, desde f[x0]):', coef)
21 print('\nTabla completa de diferencias divididas:\n', table)
22 # Evaluación en x=2.5
23 x_test = 2.5
24 y_test = newton_eval(coef, xs, x_test)
25 print(f'P_3({x_test}) =', y_test)
```

PROBLEMS OUTPUT PORTS POSTMAN CONSOLE DEBUG CONSOLE

▼ TERMINAL

```
PS C:\Users\pc-\modelado\semana10> & C:/Users/pc-/AppData/Local/Programs/Python/Python313/python.exe c:/Users/pc-/modelado/semana10/1ejercicio1.py
Coeficientes (diferencias divididas, desde f[x0]): [1. 3. 1. 0.]

Tabla completa de diferencias divididas:
[[ 1.  3.  1.  0.]
 [ 4.  5.  1.  0.]
 [ 9.  7.  0.  0.]
 [16.  0.  0.  0.]]
P_3(2.5) = 6.25
PS C:\Users\pc-\modelado\semana10>
```

2- EJERCICIO

```
1ejercicio1.py 2ejercicio2.py X 3ejercicio.py 4ejercicio.py 5ejercicio.py
2ejercicio2.py > ...
1 #EJERCICIO 2
2 import numpy as np
3 def newton_interpolation(x_data, y_data, x):
4     n = len(x_data)
5     coef = np.copy(y_data).astype(float)
6     for j in range(1, n):
7         coef[j:n] = (coef[j:n] - coef[j-1:n-1]) / (x_data[j:n] - x_data[0:n-j])
8     result = coef[-1]
9     for k in range(n-2, -1, -1):
10        result = result * (x - x_data[k]) + coef[k]
11    return result, coef
12 x_points = np.array([0, 1, 2])
13 y_points = np.array([1, 4, 9])
14 val, coef = newton_interpolation(x_points, y_points, 1.5)
15 print("Coeficientes:", coef)
16 print("P(1.5) =", val)
```

PROBLEMS OUTPUT PORTS POSTMAN CONSOLE DEBUG CONSOLE

▼ TERMINAL

```
PS C:\Users\pc-\modelado\semana10> & C:/Users/pc-/AppData/Local/Programs/Python/Python313/python.exe c:/Users/pc-/modelado/semana10/2ejercicio2.py
Coeficientes: [1. 3. 1.]
P(1.5) = 6.25
PS C:\Users\pc-\modelado\semana10>
```

3- EJERCICIO

```
3ejercicio.py > newton_interp
1  # EJERCICIO 3
2  import numpy as np
3  import math
4  x_points = np.array([0, 1, 2])
5  y_points = np.array([math.e**x for x in x_points])
6  def newton_interp(x_data, y_data, x):
7      n = len(x_data)
8      coef = np.copy(y_data)
9      for j in range(1, n):
10         coef[j:n] = (coef[j:n] - coef[j-1:n-1]) / (x_data[j:n] - x_data[0:n-j])
11     result = coef[-1]
12     for k in range(n-2, -1, -1):
13         result = result * (x - x_data[k]) + coef[k]
14     return result
15 print("P(1.5) =", newton_interp(x_points, y_points, 1.5))
```

PROBLEMS OUTPUT PORTS POSTMAN CONSOLE DEBUG CONSOLE

✓ TERMINAL

```
PS C:\Users\pc-\modelado\semana10> & C:/Users/pc-/AppData/Local/Programs/Python/Python313/Python313.exe 3ejercicio.py
P(1.5) = 4.684607408443277
PS C:\Users\pc-\modelado\semana10>
```

4- EJERCICIO

```
4ejercicio.py > ...
1  # EJERCICIO 4
2  import numpy as np
3  import math
4  x_points = np.array([0, math.pi/4, math.pi/2])
5  y_points = np.sin(x_points)
6  def newton_interp(x_data, y_data, x):
7      n = len(x_data)
8      coef = np.copy(y_data)
9      for j in range(1, n):
10         coef[j:n] = (coef[j:n] - coef[j-1:n-1]) / (x_data[j:n] - x_data[0:n-j])
11     result = coef[-1]
12     for k in range(n-2, -1, -1):
13         result = result * (x - x_data[k]) + coef[k]
14     return result
15 x_eval = math.pi / 3
16 print("P(pi/3) =", newton_interp(x_points, y_points, x_eval))
17 print("Valor real =", math.sin(x_eval))
```

PROBLEMS OUTPUT PORTS POSTMAN CONSOLE DEBUG CONSOLE

✓ TERMINAL

```
PS C:\Users\pc-\modelado\semana10> & C:/Users/pc-/AppData/Local/Programs/Python/Python313/Python313.exe 4ejercicio.py
P(pi/3) = 0.8507615832769311
Valor real = 0.8660254037844386
PS C:\Users\pc-\modelado\semana10>
```