geometria-plana-resumo

April 1, 2024

[ ]:

```python
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = [4, 3]
```

0.1 Pontos Notáveis do Triângulo

Ponto	Intersecção das três…	Nota
Baricentro	Medianas	O baricentro é o centro de massa (gravidade) do triângulo.
Incentro	Bissetrizes internas	Centro da circunferência inscrita no triângulo.
Circuncentro	Mediatrizes	Centro da circunferência circunscrita ao triângulo.
Ortocentro	Retas suportes das alturas	

0.2 Polígonos

DEFINIÇÃO. Seja uma sequência $s = (A_1, A_2, ..., A_n)$ de pontos distintos no plano com $n \geq 3$, onde três pontos consecutivos (A_{n-1}, A_n, A_{n+1}) são não colineares. Chama-se de **polígono P** a união dos segmentos $\overline{A_1A_2}$, $\overline{A_2A_3}$, … , $\overline{A_{n-1}A_n}$, $\overline{A_nA_1}$, ou seja,

$$P = \overline{A_1A_2} \ \cup \ \overline{A_2A_3} \ \cup \ ... \ \cup \ \overline{A_{n-1}A_n} \ \cup \ \overline{A_nA_1}$$

P é comumente denotado como

$$P = A_1A_2 \ldots A_n$$

Os pontos $A_1 , A_2 , \ldots , A_n$ são chamados **vértices** do polígono.

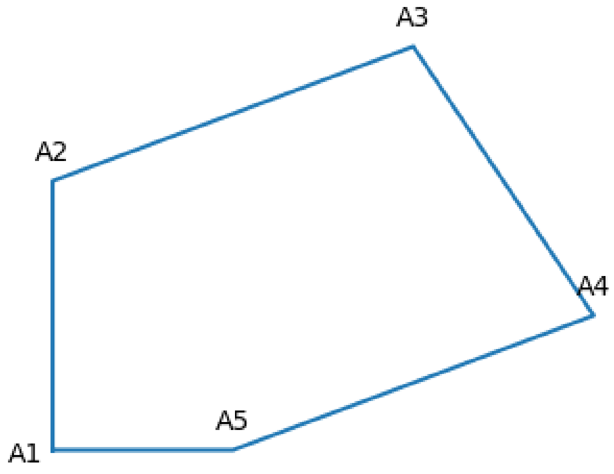
Os segmentos $A_1A_2 , A_2A_3 , \ldots , A_{n-1}A_n , A_nA_1$ são chamados **lados** do polígono.

Os ângulos $\hat{A}_1 = A_n\hat{A}_1A_2$, $\hat{A}_2 = A_1\hat{A}_2A_3$, ... , $\hat{A}_n = A_{n-1}\hat{A}_nA_1$ são os ângulos (internos) do polígono.

0.2.1 Classificação de Polígonos

Um polpigono é dito **simples** se, e somente se, a interseção de quaisquer dois lados não consecutivos é vazia.

```
[ ]: vertices = [(0, 0, 'A1'), (0, 1, 'A2'), (1, 1.5, 'A3'), (1.5, 0.5, 'A4'), (0.5,␣
      ↪0, 'A5'), (0, 0, '')]
     x_values = []
     y_values = []
     for point in vertices:
         x, y, label = point
         x_values.append(x)
         y_values.append(y)
         index = vertices.index(point)
         next_point = point
         if index < len(vertices) - 1:
              x_next, y_next, label_next = vertices[index + 1]
         plt.annotate(
             label,
             (x, y),
             textcoords='offset points',
             xytext=(-10, -4) if x == x_next else (0, 8), # distance from text to␣
      ↪points (x, y)
             ha='center'
         )
     plt.plot(x_values, y_values)
     plt.axis('off')
     plt.show()
```



```
[ ]:
```