

### Atividade: Conjuntos Numéricos e Manipulação em Python

Objetivo: Nesta atividade, vamos trabalhar os diferentes conjuntos numéricos, suas operações básicas e como manipulá-los utilizando Python. Além disso, você verá exemplos de organização e tratamento de dados numéricos.

#### Parte 1: Introdução aos Conjuntos Numéricos

Os conjuntos numéricos são categorias de números com propriedades distintas. Os principais conjuntos são:

Conjunto	Símbolo	Exemplo de Elementos
Naturais	$\mathbb{N}$	$\{0, 1, 2, 3, \dots\}$
Inteiros	$\mathbb{Z}$	$\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
Racionais	$\mathbb{Q}$	$\{\dots; -\frac{1}{2}; 0; \frac{2}{3}; 1; 3,5; \dots\}$
Reais	$\mathbb{R}$	Todos os racionais + irracionais ( $\pi, \sqrt{2}, etc.$ )
Complexos	$\mathbb{C}$	$\{a + bi \mid a, b \in \mathbb{R}, i^2 = -1\}$

#### Exemplo no Python:

```
naturais = {0, 1, 2, 3, 4, 5}
inteiros = {-3, -2, -1, 0, 1, 2, 3}
racionais = {0.5, -2.3, 1.75}
reais = {3.14, -1.414, 2.718}
complexos = {complex(1, 2), complex(-3, -4)}
print("Conjunto de números naturais:", naturais)
print("Conjunto de números inteiros:", inteiros)
print("Conjunto de números racionais:", racionais)
print("Conjunto de números reais:", reais)
print("Conjunto de números complexos:", complexos)
```

#### Parte 2: Operações Básicas entre Conjuntos (30 min)

Os conjuntos numéricos permitem operações como união, interseção, diferença e diferença simétrica.

Exemplo no Python:

```
A = {1, 2, 3, 4, 5}
B = {3, 4, 5, 6, 7}

print("União:", A | B) # A ∪ B
print("Interseção:", A & B) # A ∩ B
print("Diferença A - B:", A - B) # Elementos em A, mas não em B
print("Diferença Simétrica:", A ^ B) # Elementos exclusivos de A ou B
```

### Parte 3: Manipulação de Conjuntos Numéricos no Python (30 min)

Exemplo no Python:

```
numeros = {1, 2, 3, 4, 5}

numeros.add(6) # Adiciona um elemento
numeros.remove(2) # Remove um elemento existente
numeros.update([7, 8, 9]) # Adiciona múltiplos elementos
print("Conjunto atualizado:", numeros)
print("Tamanho do conjunto:", len(numeros))
```

### Parte 4: Organização e Tratamento de Dados Numéricos

Exemplo no Python:

```
idades = [25, 30, 22, 22, 28, 35, 40, 30, -5, 120, 25, 18, 29]

# Remover valores inválidos (idades negativas ou irreais)
idades_validas = {i for i in idades if 0 <= i <= 100}

# Estatísticas básicas
idade_min = min(idades_validas)
idade_max = max(idades_validas)
media_idades = sum(idades_validas) / len(idades_validas)

print("Idades únicas e válidas:", idades_validas)
print("Idade mínima:", idade_min)
print("Idade máxima:", idade_max)
print("Média de idade:", round(media_idades, 2))
```

### Conclusão

O que aprendemos:

1. Diferença entre os principais conjuntos numéricos.
2. Como realizar operações básicas entre conjuntos.
3. Como manipular conjuntos numéricos no Python.
4. Como tratar e organizar dados numéricos de forma eficiente.