

# CAPÍTULO 7

---

## Conjuntos

---

### 7.1 Conceitos

Conjuntos são estruturas de dados que armazenam coleções **não ordenadas** e únicas de elementos. Esses elementos podem ser de qualquer tipo, como números, strings, tuplas e até mesmo outros conjuntos.

Alguns uso de conjuntos:

- Verificação de ocorrências
- Remoção de duplicações
- Operações matemáticas: união, interseção, diferença e diferença simétrica.

### 7.2 Sintaxe e operações

Para criar um conjunto em Python, você pode usar chaves `()` ou a função embutida `set()`. Por exemplo:

```
1 conjunto_vazio = {} # ou
2 conjunto_vazio = set()
```

Em Python, você pode realizar operações matemáticas com conjuntos. As mais comuns são:

União: retorna um novo conjunto contendo todos os elementos presentes em ambos os conjuntos.

```
1 >>> uniao = conjunto1.union(conjunto2)
```

Interseção: retorna um novo conjunto contendo apenas os elementos presentes em ambos os conjuntos.

```
1 >>> intersecao = conjunto1.intersection(conjunto2)
```

Diferença: retorna um novo conjunto contendo os elementos presentes no primeiro conjunto, mas não no segundo.

```
1 >>> diferenca = conjunto1.difference(conjunto2)
```

Diferença simétrica: retorna um novo conjunto contendo os elementos presentes em um dos conjuntos, mas não em ambos.

```
1 >>> diferenca_simetrica = conjunto1.symmetric_difference(
    conjunto2)
```

## 7.3 Exemplos

Os conjuntos podem ser criados a partir de listas:

```
1 >>> gene_list = ['BRCA1', 'BRCA2', 'MUTYH', 'MLH1']
2 >>> gene_set = set(gene_list) # conversão para conjuntos
```

Adicionando e removendo elementos ao conjunto:

```
1 >>> gene_set.add("TOMM40")
2 >>> gene_set.remove("TOMM40")
```

Intersecção entre conjuntos:

```
1 >>> genes1 = {"APOE", "TOMM40", "FTO", "VDR"} # gene set 1
2 >>> genes2 = {"APOE", "FTO", "VDR"} # gene set 2
3 >>> genes_em_comum = genes1 & genes2
4 >>> print(genes_em_comum) # ou
5 >>> genes_em_comum = genes1.intersection(genes2)
6 >>> print(genes_em_comum)
```

Diferença entre conjuntos:

```
1 >>> genes1 = {"APOE", "TOMM40", "FTO", "VDR"}
2 >>> genes2 = {"APOE", "FTO", "VDR", "APOC1"}
3 >>> diferenca = genes1 - genes2
4 >>> print(diferenca)
5 {'TOMM40'}
6 >>> diferenca = genes1.difference(genes2) # alternativa
7 >>> print(diferenca)
8 {'TOMM40'}
```

Diferença simétrica entre conjuntos:

```
1 genes1 = {"APOE", "TOMM40", "FTO", "VDR"}
2 genes2 = {"APOE", "FTO", "VDR", "APOC1"}
3 diferenca_simetrica = genes1.symmetric_difference(genes2)
4 print(diferenca_simetrica)
```