

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**

# Lógica de programação e algoritmos

## Algoritmos de contagem e acumulação

Introdução aos algoritmos de contagem

Código da aula: [DADOS]ANO1C3B4S27A1

Lógica de  
programação e  
algoritmos

## Mapa da Unidade 1 Componente 4

semana

25

Prática de busca e  
ordenação

semana

27

**Você está aqui!**  
Algoritmos de  
contagem e  
acumulação

semana

28

Pilhas e filas

Lógica de  
programação e  
algoritmos

## Mapa da Unidade 1 Componente 4

**Você está aqui!**

Algoritmos de contagem e  
acumulação

**Aula 1**

Código da aula: [DADOS]ANO1C3B4S27A1

27



## Objetivos da Aula

- Introduzir os fundamentos dos Algoritmos de Contagem e Acumulação.



## Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet.



## Duração da Aula

50 minutos.



## Competências Técnicas

- Identificar e resolver problemas relacionados a dados e análises;
- Compreender e dominar técnicas de manipulação de dados.



## Competências Socioemocionais

- Adaptar-se a novas tecnologias, técnicas e tendências sem perder o foco, as metas e os objetivos da organização;
- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados; trabalhar em equipes multifuncionais colaborando com colegas, gestores e clientes.





## Primeiras ideias

Algoritmos de contagem e algoritmos de acumulação são fundamentais em análises de dados e programação, permitindo a extração de informações importantes e a realização de cálculos complexos.

Como você definiria a diferença entre contagem e acumulação em um algoritmo?

Por que é importante entender algoritmos de contagem e acumulação em programação e análise de dados?

Quais são algumas aplicações práticas de algoritmos de contagem e acumulação que vocês podem imaginar no dia a dia?



## Ponto de partida

Imagine que vocês são analistas de dados em uma empresa de comércio eletrônico.

A empresa quer analisar o desempenho das vendas durante um mês para entender quais produtos foram mais vendidos e calcular a receita total gerada. Vocês precisam criar algoritmos de contagem para determinar a frequência de vendas de cada produto e algoritmos de acumulação para calcular a receita total.

Respondam:

1. Quais estruturas de dados vocês usariam para armazenar a contagem de vendas de cada produto e por quê?
2. Como vocês lidariam com produtos que não foram vendidos durante o mês? Como isso afetaria os algoritmos de contagem e acumulação?
3. Se quisermos analisar a média de vendas diárias por produto, que modificações precisaríamos fazer nos algoritmos de acumulação?

Situação fictícia elaborada especialmente para o curso.



## Construindo o **conceito**

# O que são algoritmos de contagem?

Os algoritmos de contagem são aqueles que têm como objetivo contar a ocorrência de elementos em uma coleção de dados.

São frequentemente utilizados para determinar a frequência de itens, como:

- ✓ a contagem de palavras em um texto;
- ✓ a contagem de votos em uma eleição;
- ✓ a contagem de itens em um inventário.

## Construindo o **conceito**

# Estruturas de dados para contagem

As estruturas de dados mais comuns para contagem incluem:

- ▶ **Listas:** utilizadas para armazenar múltiplos itens em uma única variável;
- ▶ **Dicionários:** armazenam pares de chave-valor, em que a chave é o item que está sendo contado e o valor é a contagem desse item.

## Construindo o **conceito**

### Laços de repetição

Laços de repetição são fundamentais para algoritmos de contagem, pois permitem iterar sobre coleções de dados:

- ▶ **For loop:** usado para iterar sobre uma sequência (como uma lista, tupla ou *string*) e executar um bloco de código para cada item na sequência;
- ▶ **While loop:** executa um bloco de código enquanto uma condição é verdadeira.

Construindo  
o **conceito**

## Condicionais em algoritmos de contagem

Condicionais são usados para testar condições específicas durante a contagem e tomar decisões com base nesses testes:

- ▶ **If statement:** executa um bloco de código se uma condição for verdadeira.
- ▶ **Elif e Else:** permitem testar múltiplas condições.



## Construindo o **conceito**

### Exemplo 1: contagem de vogais em uma *string*

A função **contar\_vogais**:

- conta o número de vogais em uma ***string***;
- utiliza um laço **for** para iterar sobre cada caractere no texto e uma condicional **if** para verificar se o caractere é uma vogal.

```
def contar_vogais(texto):  
    vogais = 'aeiouAEIOU'  
    contagem = 0  
    for char in texto:  
        if char in vogais:  
            contagem += 1  
    return contagem  
  
texto = "Exemplo de contagem de vogais"  
print(contar_vogais(texto))
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

## Construindo o **conceito**

### Exemplo 2: contagem de palavras em um texto

A função **contar\_palavras**:

- divide o texto em palavras usando o método **split**;
- conta o número de palavras usando **len**.

```
def contar_palavras(texto):  
    palavras = texto.split()  
    contagem = len(palavras)  
    return contagem  
  
texto = "Este é um exemplo de contagem de palavras"  
print(contar_palavras(texto))
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

## Construindo o **conceito**

### Exemplo 3: contagem de caracteres específicos

A função **contar\_caracteres**:

- conta a ocorrência de um caractere específico em uma **string**;
- usa um laço **for** e uma condicional **if** para verificar cada caractere.

```
def contar_caracteres(texto, caractere):  
    contagem = 0  
    for char in texto:  
        if char == caractere:  
            contagem += 1  
    return contagem  
  
texto = "Exemplo de contagem de caracteres específicos"  
caractere = 'e'  
print(contar_caracteres(texto, caractere))
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



Vamos  
fazer um  
**quiz**

**Qual estrutura armazena pares chave-valor?**

Lista.

Tupla.

Dicionário.

Conjunto.





Vamos  
fazer um  
**quiz**

**Qual laço é usado para iterar sequências?**

*If*

*For*

*While*

*Elif*

Vamos  
fazer um  
**quiz**

Qual método separa palavras em um texto?

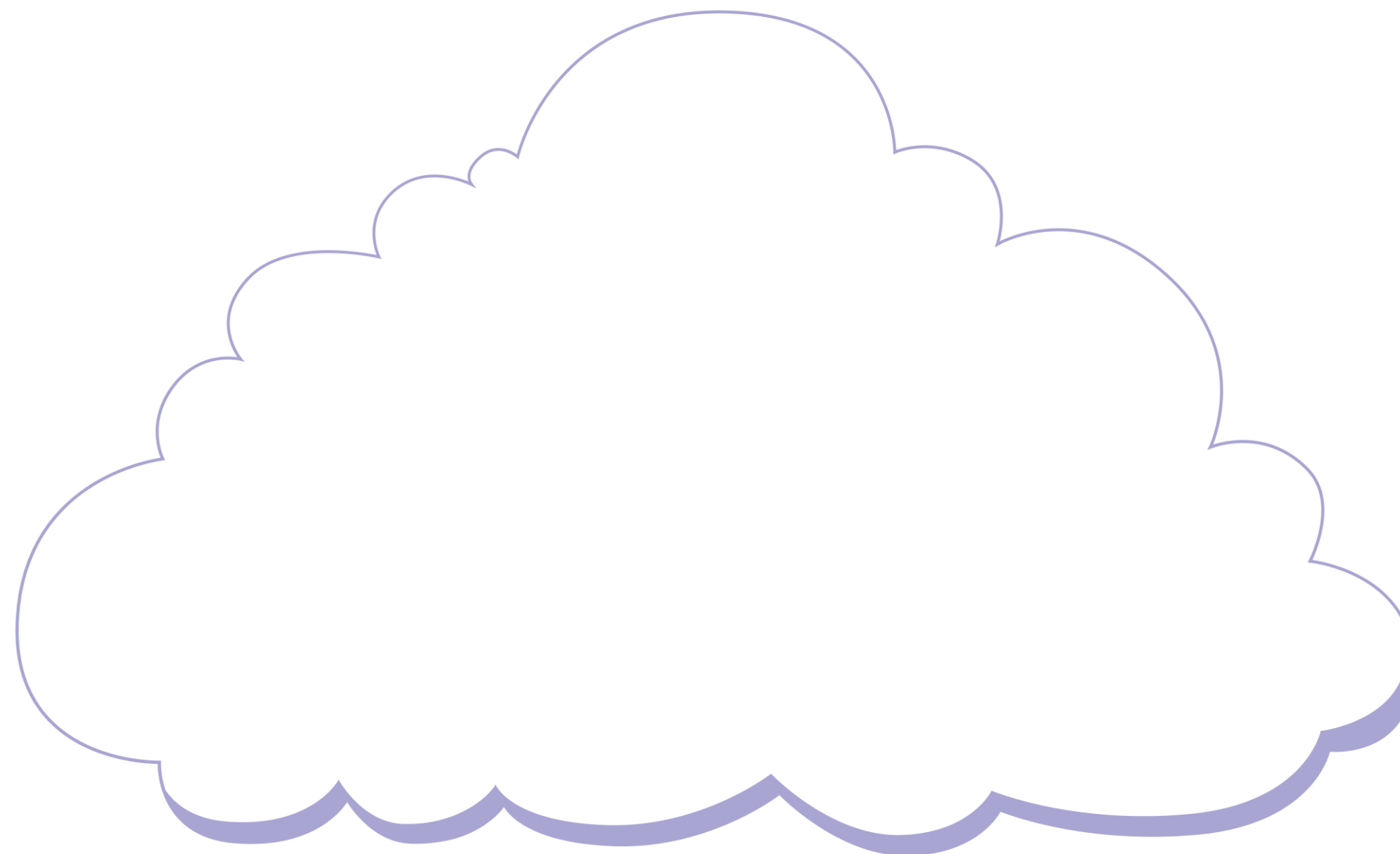
*Join*

*Replace*

*Split*

*Strip*

# Nuvem de palavras



© Getty Images

O que nós  
**aprendemos  
hoje?**





© Getty Images

O que nós  
**aprendemos  
hoje?**

## Então ficamos assim...

- 1** Abordamos algoritmos de contagem, explicando seu objetivo de contar ocorrências em coleções de dados, como textos ou inventários. Esses algoritmos são fundamentais para determinar a frequência de itens em diversos contextos;
- 2** Foram abordadas estruturas de dados essenciais, como listas e dicionários, utilizadas para armazenar elementos e suas contagens, e também os laços de repetição, como *for* e *while*, e condicionais, como *if* e *else*, como ferramentas importantes nesses algoritmos;
- 3** Os exemplos práticos incluíram a contagem de vogais em uma *string*, palavras em um texto e caracteres específicos, demonstrando a aplicação de algoritmos de contagem. Foram utilizadas as estruturas de dados e laços de repetição para alcançar resultados precisos.



# Saiba mais

Precisa organizar seus contatos telefônicos em Python de forma eficiente? Um dicionário pode ser a solução ideal!

ORESTES, Y. Python: trabalhando com dicionários. Alura, 16 out. 2018. Disponível em: <https://www.alura.com.br/artigos/trabalhando-com-o-dicionario-no-python/>. Acesso em: 16 jul. 2024.

# Referências da aula

Identidade visual: imagens © Getty Images

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python. Porto Alegre: Bookman, 2022.

ORESTES, Y. Python: trabalhando com dicionários. Alura, 16 out. 2018. Disponível em: <https://www.alura.com.br/artigos/trabalhando-com-o-dicionario-no-python/> . Acesso em: 16 jul. 2024.

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**