

Educação Profissional Paulista

Técnico em
**Ciência de
Dados**

Lógica de programação e algoritmos

Algoritmos de contagem e acumulação

Acumulação de valores

Código da aula: [DADOS]ANO1C3B4S27A2

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 2

Código da aula: [DADOS]ANO1C3B4S27A2

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.

Construindo o **conceito**

Algoritmos de acumulação

Algoritmos de acumulação são aqueles que agregam valores ao longo de uma coleção de dados. São usados para somar valores, calcular médias ou mesmo combinar strings.

Variáveis acumuladoras

Variáveis acumuladoras são usadas para manter um total corrente ao longo da execução de um laço. Inicialmente, são definidas com um valor neutro (como 0 para soma).

Algoritmos de acumulação

Acumulação com estruturas de dados

- Listas: podem armazenar múltiplos valores acumulados.
- Dicionários: podem armazenar acumuladores separados para diferentes categorias ou chaves.

Laços de repetição para acumulação

Laços como for e while são usados para iterar sobre coleções e atualizar a variável acumuladora em cada iteração.

Construindo o **conceito**

Exemplo 1: soma de elementos de uma lista

A função **somar_elementos** soma todos os elementos de uma lista, utilizando um laço **for** para iterar sobre cada número e acumulando o total.

```
def somar_elementos(lista):  
    total = 0  
    for numero in lista:  
        total += numero  
    return total  
  
numeros = [1, 2, 3, 4, 5]  
print(somar_elementos(numeros))
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Exemplo 2: acumulação de produto de elementos

A função **produto_elementos** calcula o produto de todos os elementos de uma lista, acumulando o valor em cada iteração do laço **for**.

```
def produto_elementos(lista):  
    produto = 1  
    for numero in lista:  
        produto *= numero  
    return produto  
  
numeros = [1, 2, 3, 4, 5]  
print(produto_elementos(numeros))
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplo 3: acumulação de *strings*

A função **concatenar_strings** acumula *strings* em uma única *string*, usando um laço **for** para iterar sobre a lista de *strings* e acumulando o resultado.

```
def concatenar_strings(lista):  
    resultado = ""  
    for string in lista:  
        resultado += string + " "  
    return resultado.strip()  
  
strings = ["Este", "é", "um", "exemplo"]  
print(concatenar_strings(strings))
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



Colocando em **prática**

Exercício: análise de vendas de produtos

Crie um programa em Python que analise um registro de vendas de produtos. O programa deve contar a frequência de vendas de cada produto e calcular a receita total acumulada para cada um.

Use estruturas de dados apropriadas, laços de repetição e condicionais para realizar essas tarefas.

Tarefas:

1. Crie um dicionário para armazenar a contagem de vendas de cada produto.
2. Crie outro dicionário para armazenar a receita total acumulada de cada produto.
3. Para cada venda registrada, atualize a contagem de vendas e a receita acumulada.
4. No final, exiba a contagem de vendas e a receita total para cada produto.



20 minutos



Duplas



**Código de
programação**



Colocando em **prática**

Exercício: análise de vendas de produtos

Entrada:

Uma lista de tuplas, em que cada tupla contém o nome do produto, a quantidade vendida e o preço unitário. Use a tabela abaixo na sua solução!

```
vendas = [  
    ("Produto A", 10, 5.0),  
    ("Produto B", 5, 12.0),  
    ("Produto A", 3, 5.0),  
    ("Produto C", 8, 7.5),  
    ("Produto B", 7, 12.0)  
]
```



20 minutos



Dúplas



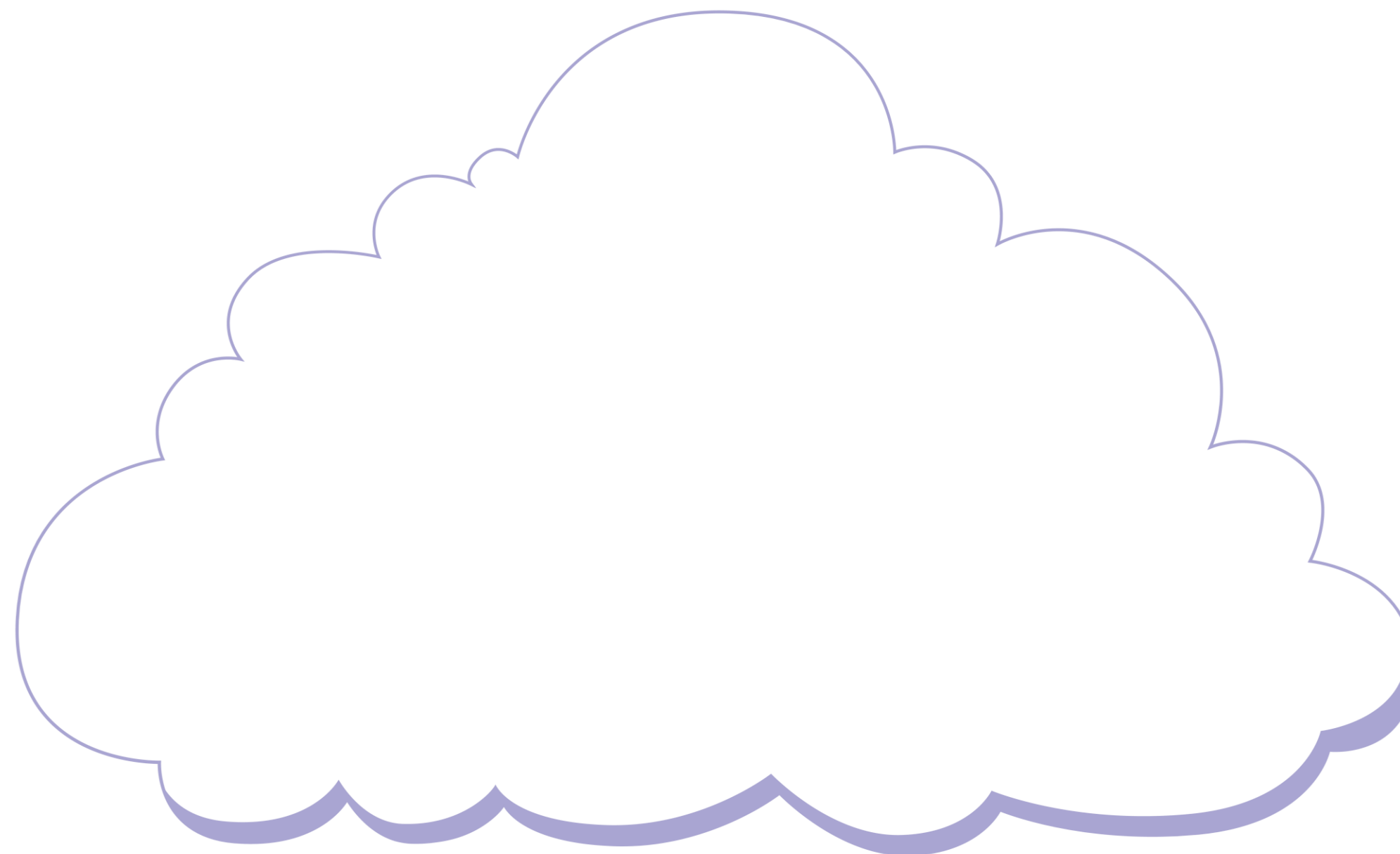
Código de
programação

Saída:

Exibir a contagem de vendas e a receita acumulada para cada produto.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Nuvem de palavras



© Getty Images

O que nós
**aprendemos
hoje?**



© Getty Images

O que nós
**aprendemos
hoje?**

Então ficamos assim...

- 1** Abordamos algoritmos de acumulação, explicando como acumulam valores em uma coleção de dados. São usados para somar, calcular médias e combinar *strings*, sendo essenciais para análises e cálculos;
- 2** As variáveis acumuladoras foram discutidas como fundamentais para manter um total corrente durante a execução de laços de repetição, como *for* e *while*. Essas variáveis começam com um valor neutro, como zero para somas;
- 3** Os exemplos práticos incluíram a soma de elementos em uma lista, o cálculo do produto de elementos e a concatenação de *strings*. Esses exemplos ilustraram como utilizar laços e variáveis acumuladoras em diferentes contextos.

Saiba mais

Afinal, qual a diferença entre as tuplas e as listas na linguagem Python? Você sabe quando utilizar uma ou outra?

ORESTES, Y. Tupla no Python: o que é, como criar e manipular e suas diferenças com as listas. Alura, 2 mar. 2023. Disponível em: <https://www.alura.com.br/artigos/conhecendo-as-tuplas-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. Tupla no Python: o que é, como criar e manipular e suas diferenças com as listas. Alura, 2 mar. 2023. Disponível em: <https://www.alura.com.br/artigos/conhecendo-as-tuplas-no-python/>. Acesso em: 16 jul. 2024.

Educação Profissional Paulista

Técnico em
**Ciência de
Dados**