

1. Estrutura de dados inicial

Vamos começar com os dados das vendas.

Cada item da lista representa uma venda contendo três informações: produto, quantidade vendida e preço unitário.

python

```
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)  
]
```

2. Preparando os acumuladores

Precisamos criar dois dicionários vazios para armazenar os resultados:

python

```
contagem_vendas = {}  
receita_total = {}
```

contagem_vendas : vai guardar o total de unidades vendidas por produto.

receita_total : vai guardar o total de dinheiro arrecadado por produto.

3. Criando o laço de repetição

O próximo passo é percorrer a lista de vendas e atualizar os valores de contagem e receita.

python

```
for produto, quantidade, preco in vendas:  
    if produto in contagem_vendas:  
        contagem_vendas[produto] += quantidade  
        receita_total[produto] += quantidade * preco  
    else:  
        contagem_vendas[produto] = quantidade  
        receita_total[produto] = quantidade * preco
```

O que acontece aqui?

O **for** percorre todos os itens da lista **vendas** .

A variável **produto** é o nome do produto (ex: "Produto A").

A variável **quantidade** mostra quantas unidades foram vendidas.

A variável **preco** indica o valor unitário.

Se o produto já existir no dicionário, o código **atualiza** os valores.

Se for a **primeira vez**, os valores são criados.

4. Exibindo os resultados

Agora que os cálculos estão prontos, podemos imprimir o relatório de vendas:

python

```
print("Relatório de vendas:\n")

for produto in contagem_vendas:
    print(f"{produto}: {contagem_vendas[produto]} unidades vendidas - "
          f"Receita Total: R$ {receita_total[produto]:.2f}")
```

Saída esperada:

text

Relatório de vendas:

```
Produto A: 13 unidades vendidas - Receita Total: R$ 65.00
Produto B: 12 unidades vendidas - Receita Total: R$ 144.00
Produto C: 8 unidades vendidas - Receita Total: R$ 60.00
```

5. Tornando o código mais organizado

Podemos transformar nosso código em uma função reutilizável:

python

```
def analisar_vendas(vendas):
    contagem_vendas = {}
    receita_total = {}

    for produto, quantidade, preco in vendas:
        if produto in contagem_vendas:
            contagem_vendas[produto] += quantidade
            receita_total[produto] += quantidade * preco
        else:
            contagem_vendas[produto] = quantidade
            receita_total[produto] = quantidade * preco

    for produto in contagem_vendas:
        print(f"{produto}: {contagem_vendas[produto]} unidades - "
              f"Receita: R$ {receita_total[produto]:.2f}")
```

Para rodar o programa:

python

```
analisar_vendas(vendas)
```

6. Desafio

- . Adicione mais produtos à lista **vendas** e veja se o código continua funcionando.
- . Modifique o programa para calcular a média de preço por produto.
- . Crie um relatório que exiba o produto mais vendido.

Resumo do que você aprendeu

Como percorrer listas de registros usando loops **for** .

Como usar dicionários para armazenar e atualizar informações.

A lógica dos acumuladores e contadores em algoritmos.

Como montar um relatório final com os resultados.