Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

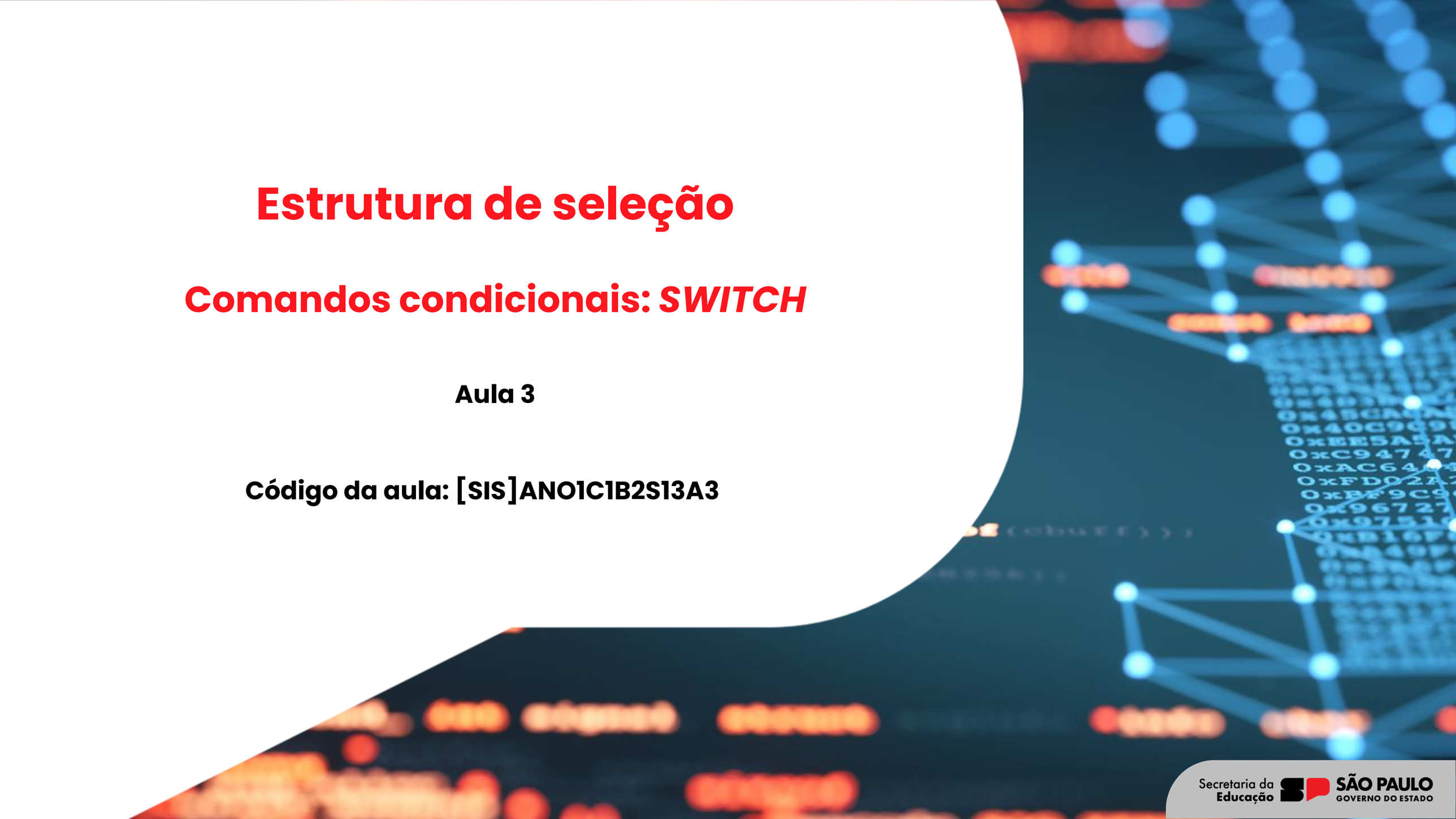
SÃO PAULO
GOVERNO DO ESTADO

Estrutura de seleção

Comandos condicionais: *SWITCH*

Aula 3

Código da aula: [SIS]ANO1C1B2S13A3



Exposição

Objetivos da Aula

- Compreender as boas práticas durante a aplicação do comando SWITCH no contexto de desenvolvimento de software.

Competências da Unidade (Técnicas e Socioemocionais)

- Desenvolver sistemas computacionais, utilizando ambiente de desenvolvimento.
- Trabalhar a criatividade e a resolução de problemas computacionais.

Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Folhas sulfite, canetas coloridas e lápis.

Duração da Aula

50 minutos

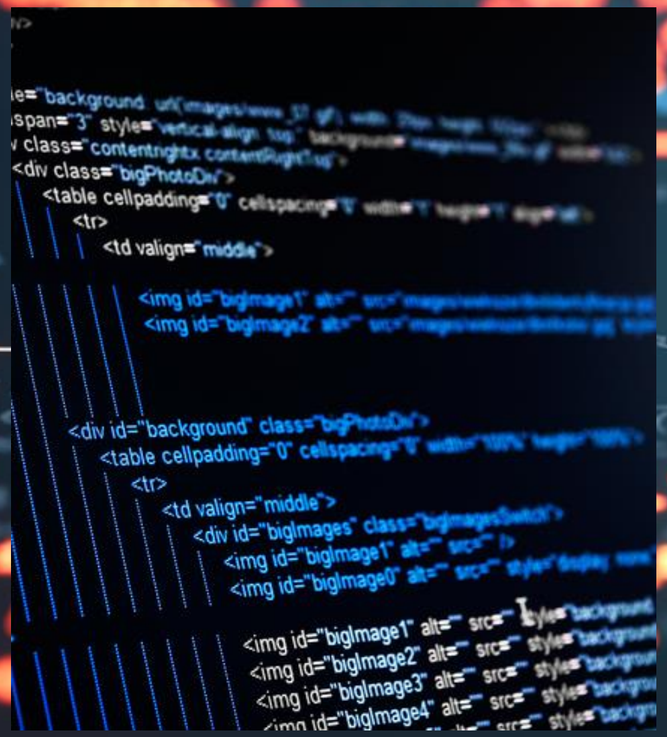
Exposição

Boas práticas e armadilhas com *SWITCH*

Foco **nas melhores práticas para usar o SWITCH de forma eficaz e segura, evitando erros comuns.**

- Boas práticas no uso do SWITCH.
- Armadilhas e erros comuns com SWITCH.
- **Refatoração de código:** melhorando o uso do SWITCH.

Exposição



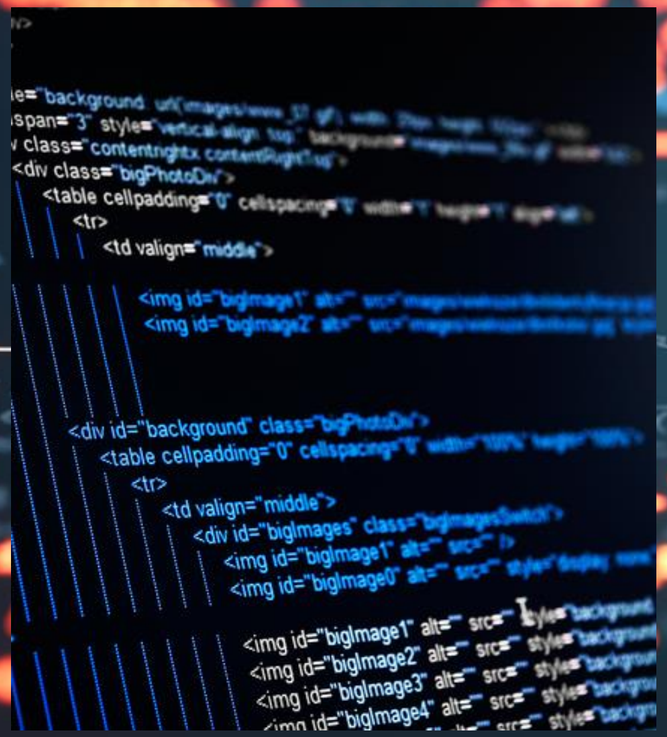
© Getty Images

Boas práticas no uso do *SWITCH*

Conforme comentado em outras aulas, **Python não tem um comando SWITCH nativo, como outras linguagens (por exemplo, C ou Java).**

No entanto, é comum emular um SWITCH, usando um dicionário ou estruturas condicionais (if-elif-else).

Exposição



© Getty Images

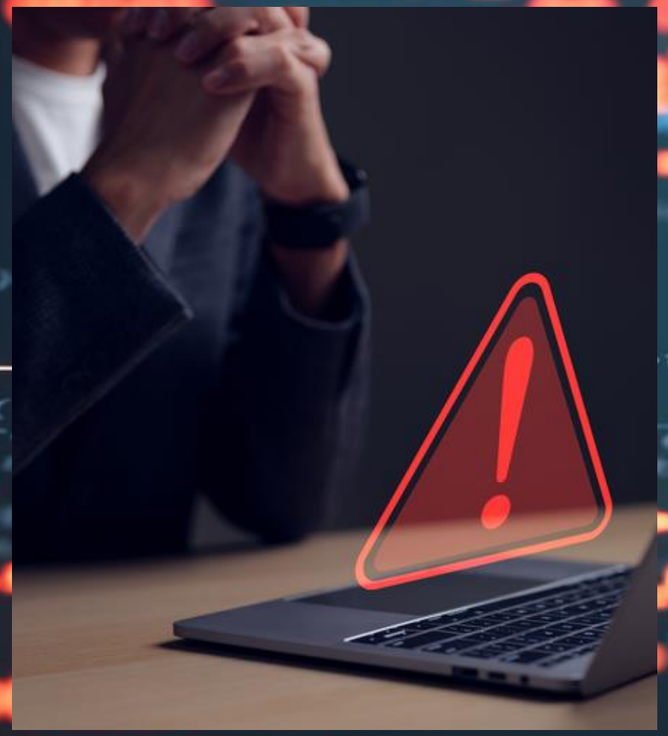
Exemplo

Suponha que você **esteja construindo um sistema para uma cafeteria para calcular o preço de diferentes tipos de café:**

```
def calcular_preco(cafe):
    precos = {
        "espresso": 1.50,
        "latte": 2.00,
        "cappuccino": 2.25
    }
    return precos.get(cafe, "Café não disponível")

preco = calcular_preco("espresso")
print(preco)
```

Exposição



© Getty Images

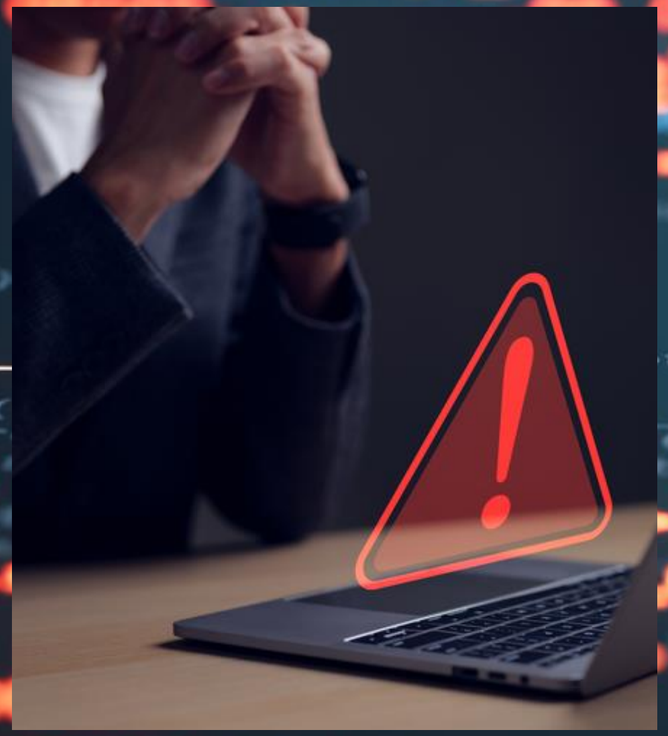
Armadilhas e erros comuns com *SWITCH*

Ausência de valor padrão: sempre forneça um valor padrão para lidar com entradas inesperadas.

Uso inadequado para lógica complexa: se a lógica associada a cada caso é complexa, é melhor usar *if-elif-else.*

Mutabilidade dos dicionários: tenha cuidado ao modificar o dicionário durante a execução.

Exposição



© Getty Images

Refatoração de código: melhorando o uso do *SWITCH*

Suponha que você tem um código com muitas condicionais e quer **simplificá-lo usando a abordagem de SWITCH com dicionário.**

Exemplo original:

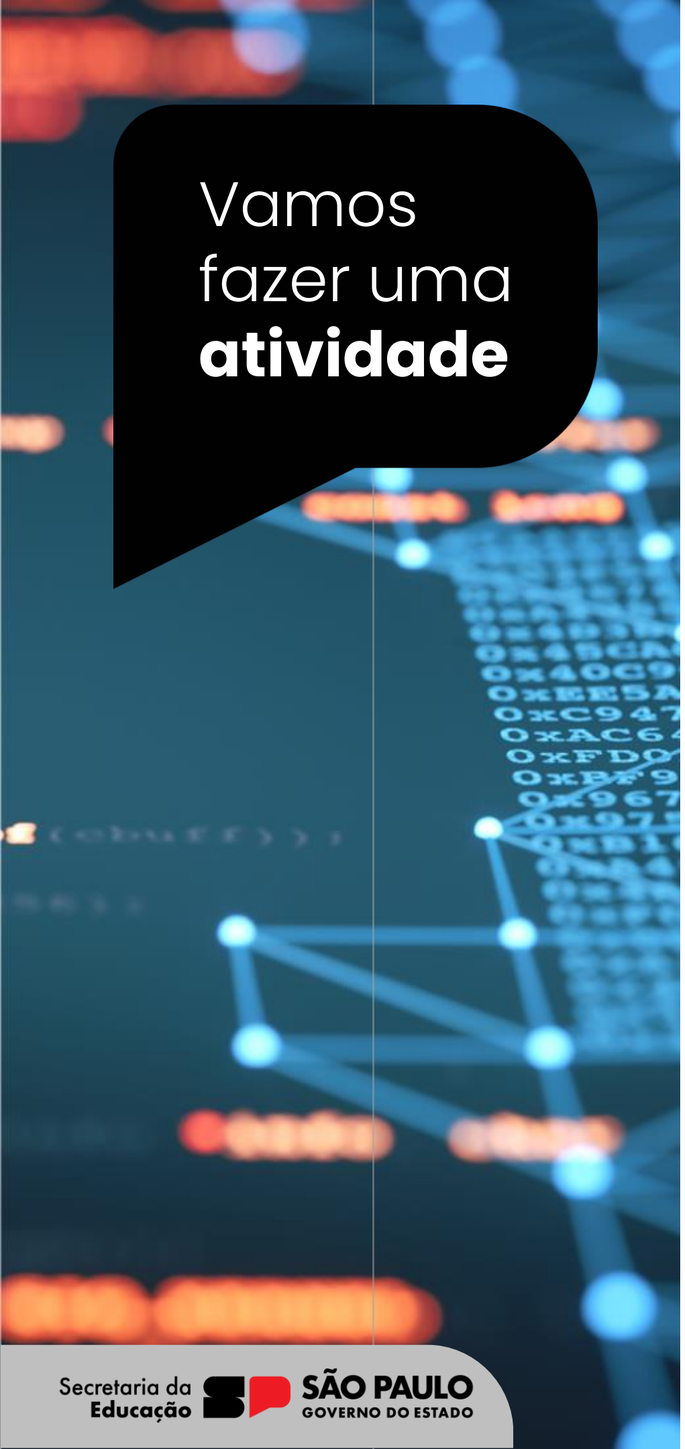
```
def aplicar_desconto(tipo_cliente, valor):
    if tipo_cliente == "estudante":
        return valor * 0.9
    elif tipo_cliente == "membro":
        return valor * 0.85
    elif tipo_cliente == "vip":
        return valor * 0.8
    else:
        return valor

valor_descontado =
aplicar_desconto("membro", 100)
print(valor_descontado
```

Após refatoração:

```
def aplicar_desconto(tipo_cliente, valor):
    descontos = {
        "estudante": lambda v: v * 0.9,
        "membro": lambda v: v * 0.85,
        "vip": lambda v: v * 0.8
    }
    return descontos.get(tipo_cliente,
lambda v: v)(valor)

valor_descontado =
aplicar_desconto("membro", 100)
print(valor_descontado)
```



Desenvolvimento de fluxograma

Objetivo: analisar e desenvolver um fluxograma para um sistema de biblioteca que categoriza livros e calcula multas de atraso, utilizando a estrutura de SWITCH (emulado via dicionário em Python).

Descrição da atividade

- **Análise da situação:**
  - A biblioteca tem três categorias de livros: "ficção", "não ficção" e "referência".
  - As multas são aplicadas de forma diferente para cada categoria: ficção a R$ 0,50 por dia, não ficção a R$ 0,60 por dia, e livros de referência não têm multa.

- **Identificação de cenários:**
  - Livro devolvido no prazo (sem multa).
  - Livro de ficção devolvido com atraso.
  - Livro de não ficção devolvido com atraso.
  - Livro de referência devolvido com atraso.

- **Objetivos do sistema:**
  - Categorizar os livros adequadamente.
  - Calcular a multa com base na categoria e nos dias de atraso.

- **Construção do fluxograma:**
  - O fluxograma deve representar o processo de tomada de decisão para determinar a multa com base na categoria do livro e no tempo de atraso.

Vamos fazer uma **atividade**

Analisar e desenvolver um fluxograma para um sistema de biblioteca que categoriza livros e calcula multas de atraso, utilizando a estrutura de *SWITCH* (emulado via dicionário em *Python*).

20 minutos

Em grupos de até cinco pessoas

Desenvolvimento de fluxograma

Exemplo

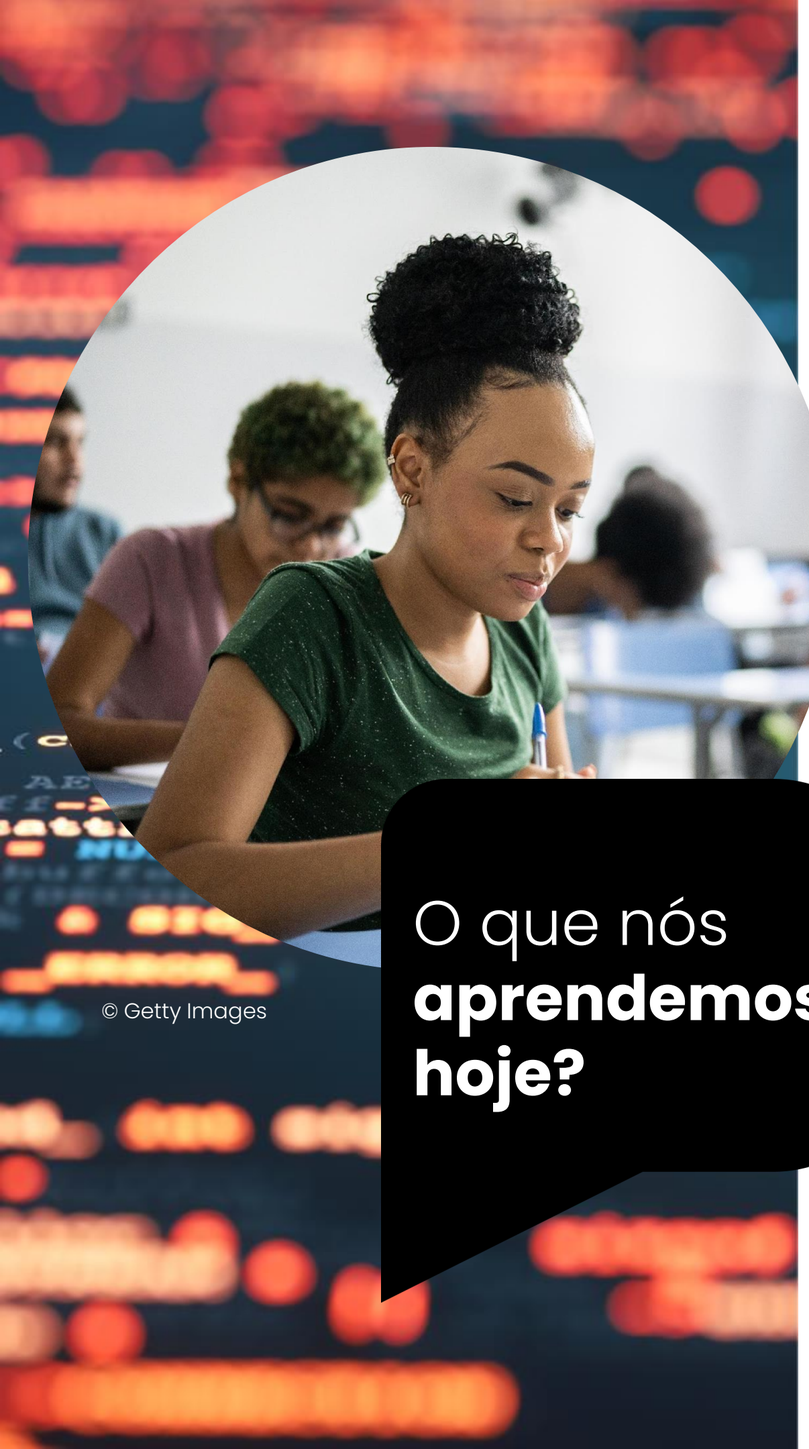
1 **Início:** comece o processo.
Recebe informações: tipo do livro e dias de atraso.

2 **Verifica categoria:**
- Se "ficção", vai para o passo 4.
- Se "não ficção", vai para o passo 5.
- Se "referência", vai para o passo 6.
- Calcula multa (ficção): multa = dias de atraso * R$ 0,50.

Retorna o valor.

3 **Calcula multa (não ficção):**
- Multa = dias de atraso * R$ 0,60. Retorna o valor.
- Sem multa (referência): retorna R$ 0,00.

Fim: processo concluído.



© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. A compreensão sobre **técnicas e boas práticas no uso do SWITCH;**

2. O conhecimento sobre alguns **exemplos de armadilhas e erros comuns com SWITCH;**

3. Na prática, o **contexto de refatoração de código: melhorando o uso do SWITCH.**

Saiba mais

Vamos aprofundar um pouco mais em **Python para interpretar as estruturas de seleção e suas principais aplicações:**

KITAMURA, C. *If-Elif-Else – Estrutura de Decisão em Python*. Disponível em: https://www.youtube.com/watch?v=3fv05eGgRIA. Acesso em: 21 mar. 2024.

Referências da aula

KITAMURA, C. *If-Elif-Else – Estrutura de Decisão em Python*. Disponível em: https://www.youtube.com/watch?v=3fv05eGgRIA. Acesso em: 21 mar. 2024.

VALE, J. C. S. *Estruturas de Seleção em Python* – #07. DEV, 2022. Disponível em: https://dev.to/jcarlosvale/estruturas-de-selecao-em-python-07-2mac. Acesso em: 21 mar. 2024.

Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO