

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Estruturas de decisão compostas

Aninhamento de estruturas de decisão

Aula 3

[SIS]ANO1C1B2S12A3

Exposição



Objetivos da Aula

- Conhecer **estruturas de decisão compostas** por meio do fluxo de execução de programas;
- Compreender exemplos de **aninhamentos em estruturas de decisão**.



Competências da Unidade (Técnicas e Socioemocionais)

- Desenvolver **sistemas computacionais** utilizando ambientes de desenvolvimento;
- Migrar sistemas, implementando **rotinas e estruturas de dados mais eficazes**;
- Trabalhar a **curiosidade e a resiliência** em resolução de problemas computacionais.



Recursos Didáticos

- Recursos audiovisuais para exibição de vídeos e imagens;
- Caderno, canetas e lápis.



Duração da Aula

50 minutos

Desenvolvimento da aula

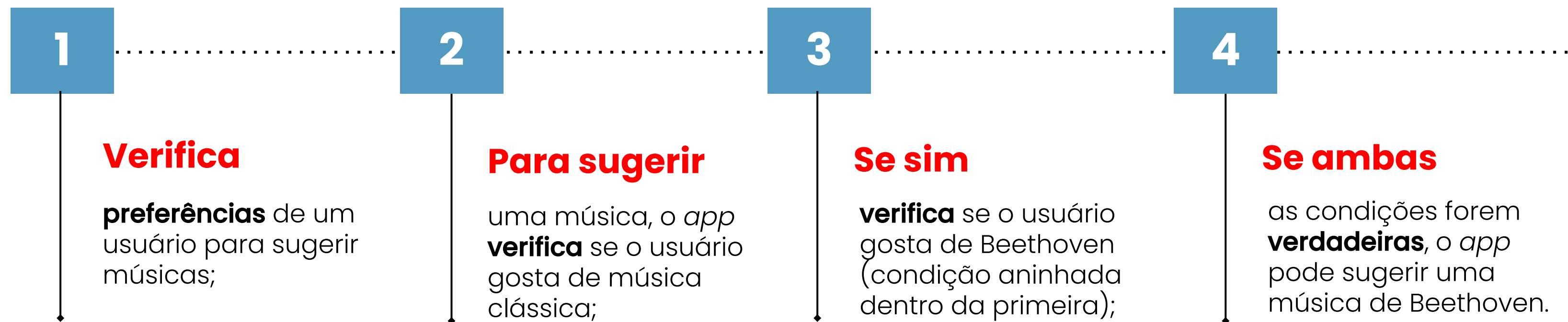
- ✓ Conceito e utilidade do **aninhamento de estruturas de decisão;**
- ✓ Análise do **fluxo de execução de um programa com estruturas de decisão aninhadas;**
- ✓ Habilidade de **escrever programas com estruturas de decisão aninhadas.**

Exposição

Aninhamento em diferentes cenários

É útil em cenários que requerem **várias verificações** e **condições**.

Exemplo de aplicativo de *streaming* de música:



Exemplo – *Python*

```
gosta_de_musica_classica = True
gosta_de_beethoven = True

if gosta_de_musica_classica:
    if gosta_de_beethoven:
        print("Sugerindo uma música de Beethoven!")
    else:
        print("Sugerindo uma música clássica.")
else:
    print("Sugerindo uma música de um gênero diferente.")
```

Exemplo – *Python* – Utilizando *and*

```
gosta_de_musica_classica = True
gosta_de_beethoven = True

if gosta_de_musica_classica and
gosta_de_beethoven:
    print("Sugerindo uma música de Beethoven!")
elif gosta_de_musica_classica:
    print("Sugerindo uma música clássica.")
else:
    print("Sugerindo uma música de um gênero
diferente.")
```

Exposição

Aninhamento de estruturas de decisão

Vamos supor que estamos criando um sistema de controle climático para uma casa.

O sistema **verifica** se é **dia** ou **noite** (usando um sensor de luz) e, então, **verifica** a **temperatura** para **decidir** se deve ligar o **ar-condicionado**, o **aquecedor** ou **nenhum** deles.



© Getty Images

Exposição

Resolução

Vamos supor que estamos criando um sistema de controle climático para uma casa.

O sistema **verifica** se é **dia** ou **noite** (usando um sensor de luz) e, então, **verifica** a **temperatura** para **decidir** se deve ligar o **ar-condicionado**, o **aquecedor** ou **nenhum** deles.

```
é_dia = True
temperatura = 22 # Em graus Celsius

if é_dia:
    if temperatura > 25:
        print("Ligando o ar-condicionado.")
    elif temperatura < 20:
        print("Ligando o aquecedor.")
    else:
        print("Temperatura confortável. Não fazendo nada.")
else:
    print("É noite. Controlador de clima em modo de economia de energia.")
```

Exposição

Casos de borda

São situações que ocorrem nos **limites extremos** das **condições** que estamos **verificando**.

Voltando ao exemplo anterior, um caso de borda pode ocorrer quando a temperatura é exatamente 25°C durante o dia.

Assim, nosso sistema atual não faria **nada**, mas talvez queiramos que o ar-condicionado seja ligado para evitar que a casa fique muito quente.



Atenção!

Em cenários complexos com múltiplas condições e estruturas de decisão aninhadas, os casos de borda podem se tornar mais complexos.

Casos de borda em cenários complexos

```
é_dia = True
temperatura = 25 # Caso de borda

if é_dia:
    if temperatura >= 25: # Modificamos a condição aqui
        para lidar com o caso de borda
        print("Ligando o ar-condicionado.")
    elif temperatura < 20:
        print("Ligando o aquecedor.")
    else:
        print("Temperatura confortável. Não fazendo nada.")
else:
    print("É noite. Controlador de clima em modo de
    economia de energia.")
```


Vamos
fazer uma
atividade

Estudo de caso

Você está programando um *app* de entrega de alimentos. O *app* deve **calcular** o **custo de entrega** com base nas seguintes **condições**:

- Se o **pedido** é **inferior a R\$10**, há uma **taxa** de entrega **fixa** de **R\$5**;
- Se o **pedido** é de **R\$10 ou mais**, mas **menos de R\$20**, a **taxa** de entrega é de **R\$3**;
- Se o **pedido** é de **R\$20 ou mais**, a **entrega** é **gratuita**.
- Se usuário é **membro** do programa de fidelidade do *app*, ele tem **desconto de R\$2** em **qualquer taxa** de entrega **aplicável**.

Escreva o algoritmo de um programa que pegue o **valor do pedido** e a informação de que o **usuário** é ou não **membro** do programa de fidelidade e, então, **calcule** e **imprima a taxa** de entrega final.

Vamos
fazer uma
atividade

Estudo de caso – Resolução Sugerida

```
valor_pedido = float(input("Digite o valor do pedido: R$"))
eh_membro_fidelidade = input("É membro do programa de fidelidade?
(sim/não): ").lower() == 'sim'
taxa_entrega = 0

# Determinando a taxa de entrega com base no valor do pedido
if valor_pedido < 10:
    taxa_entrega = 5
elif 10 <= valor_pedido < 20:
    taxa_entrega = 3
elif valor_pedido >= 20:
    taxa_entrega = 0

# Aplicando desconto para membros de fidelidade
if eh_membro_fidelidade:
    taxa_entrega = max(taxa_entrega - 2, 0) # Evita taxas negativas

print(f"A taxa de entrega é: R${taxa_entrega}")
```

Vamos
fazer uma
atividade

Estudo de caso – Resolução sugerida

Coleta de informações do usuário:

`valor_pedido`: o programa inicia pedindo ao usuário para inserir o valor do pedido. Esse valor é lido e convertido em um número de ponto flutuante (*float*) para permitir valores decimais.

`eh_membro_fidelidade`: em seguida, o programa pergunta se o usuário é membro de um programa de fidelidade. A resposta é convertida para letras minúsculas (com `.lower()`) e comparada com a *string* 'sim'. Se a resposta for 'sim', a variável `eh_membro_fidelidade` se torna *true*, caso contrário, será *false*.

Inicialização da variável `taxa_entrega`:

`taxa_entrega` é inicializada com o valor 0. Essa variável é usada para armazenar a taxa de entrega que será calculada com base no valor do pedido e no status de membro de fidelidade.

Cálculo da taxa de entrega com base no valor do pedido:

O programa, então, verifica o valor do pedido para determinar a taxa de entrega inicial:

Se `valor_pedido` for menor que 10, a taxa de entrega é definida como 5;

Se `valor_pedido` estiver entre 10 e 20 (inclusive 10, mas excluindo 20), a taxa é 3;

Se `valor_pedido` for igual ou superior a 20, não há taxa de entrega, portanto, permanece 0.

Aplicação de desconto para membros de fidelidade:

Se o usuário for membro do programa de fidelidade (`eh_membro_fidelidade` é *true*), o programa aplica um desconto de R\$2 na taxa de entrega. Isso é feito pela expressão `taxa_entrega - 2`.

No entanto, para evitar que a taxa de entrega se torne negativa, usamos a função *max* para garantir que a taxa seja, no mínimo, 0. Ou seja, `max(taxa_entrega - 2, 0)` retorna o valor da `taxa_entrega` menos 2, a menos que esse valor seja menor que 0, caso em que retorna 0.

Exibição do resultado:

Finalmente, o programa exibe a taxa de entrega calculada usando *print*, formatando a mensagem para incluir a taxa de entrega calculada.



Vamos
fazer um
quiz

Ao preparar um chá, você verifica a temperatura da água. Se a temperatura for igual a 100 graus, você desliga o fogão, senão, você continua a aquecer a água. Isso é um exemplo de que estrutura de decisão?

While

For

If/Else

Switch



Vamos
fazer um
quiz

Ao preparar um chá, você verifica a temperatura da água. Se a temperatura for igual a 100 graus, você desliga o fogão, senão, você continua a aquecer a água. Isso é um exemplo de que estrutura de decisão?

While

For

✓ *If/Else*

Switch

RESPOSTA CORRETA!

O termo *if* representa a primeira decisão, enquanto o *else*, a consequência. Nesse caso, avaliamos a tomada de decisão com estrutura simples.

Hoje desenvolvemos:

- 1** A compreensão sobre a aplicação do conceito de **estruturas de decisão aninhadas** em diferentes cenários utilizados no mundo real;
- 2** A aprendizagem sobre o conceito de **casos de borda** a partir de exposição e exemplos envolvendo esse conceito;
- 3** E trabalhamos a **construção de um algoritmo** a partir dos conceitos que aprendemos na aula de hoje.

O que nós
**aprendemos
hoje?**

© Getty Images



Saiba mais

Você sabia que o **conceito de dados mais rápidos na borda** pode ser decifrado utilizando até mesmo **situações cotidianas**?

Recomendamos esse vídeo muito bem-humorado sobre a utilização desse conceito:

LENOVO BRASIL. *Edge computing*: descomplicando o processamento de dados mais rápido nas bordas, [s.d.]. Disponível em: <https://www.youtube.com/watch?v=wLijfVmYbDw> Acesso em: 10 mar. 2024.

Referências da aula

LENOVO BRASIL. *Edge computing*: descomplicando o processamento de dados mais rápido nas bordas , [s.d.]. Disponível em: <https://www.youtube.com/watch?v=wLijfVmYbDw>. Acesso em: 10 mar. 2024.

REIS, F. Estrutura de decisão condicional composta if ... else em Python. *Bóson treinamentos em Ciência e Tecnologia*, 2022. Disponível em: <https://www.bosontreinamentos.com.br/programacao-em-python/10-python-estrutura-de-decisao-condicional-composta-se-entao-senao/#:~:text=Em%20Python%2C%20outra%20estrutura%20de,a%20condi%C3%A7%C3%A3o%20n%C3%A3o%20for%20atendida>. Acesso em: 10 mar. 2024.

Identidade visual: Imagens © Getty Images

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**