Educação Profissional Paulista

Técnico em
De sen volvim en to
de Sistem a s



Estruturas de decisão simples

Comandos condicionais

Aula 3

[SIS]ANOIC1B2S11A3





Objetivos da Aula:

Compreender como os operadores lógicos são utilizados em estruturas com condições;

Conhecer a estrutura de aplicação das condições.



Competências da Unidade (Técnicas e Socioemocionais):

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



Recursos Didáticos:

- Recurso audiovisual para exibição de vídeos e imagens;
- Caderno, canetas, lápis.



Duração da Aula:

50 minutos



Desenvolvimento da aula

Além das estruturas de decisão, outra estrutura que ajuda muito quando o assunto é o desenvolvimento de programas lógicos é a de operadores lógicos. São ferramentas essenciais para a tomada de decisões.



Operadores lógicos e sua aplicação condicional;

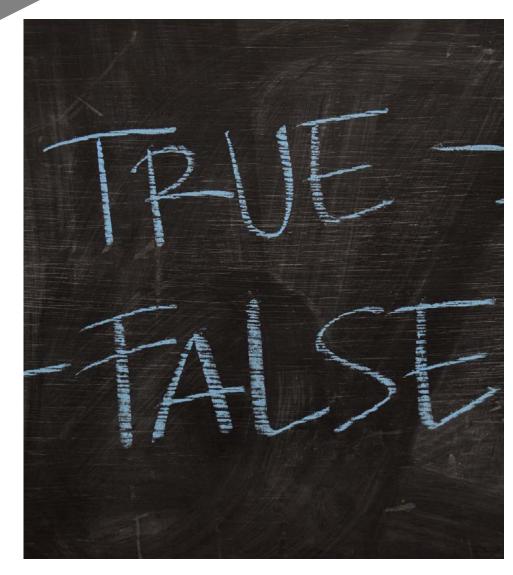


Combinação de condições utilizando operadores lógicos;



Atividades demonstrativas com códigos.





© Getty Images

Operadores lógicos

Operadores lógicos, também conhecidos como **operadores booleanos**, são fundamentais em qualquer linguagem de programação:

- eles permitem manipular valores verdadeiros (True) e falsos (False), que são a base da lógica de programação;
- eles são usados, principalmente, em instruções condicionais para combinar ou modificar condições;
- em Python, os mais usados são **and, or** e **not**.

Principais operadores lógicos

Em Python, os operadores lógicos mais usados são and, or e not.

and (e)

Esse operador retorna True se todas as condições especificadas forem verdadeiras. Caso contrário, ele retorna False.

Exemplo: na condição (5 > 3) and (6 > 3), ambas as condições são verdadeiras, então a expressão inteira é verdadeira.

or (ou)

Esse operador retorna True se pelo menos uma das condições especificadas for verdadeira. Caso contrário, ele retorna False.

Exemplo: na condição (5 < 3) or (6 > 3), a primeira condição é falsa, mas a segunda é verdadeira, então a expressão inteira é verdadeira.



Principais operadores lógicos

Em Python, os operadores lógicos mais usados são **and, or** e **not**.

not (não)

O operador lógico not é usado para negar o valor de uma expressão booleana

Ex: x = True resultado = not x print(resultado) # Isso imprimirá: False

Tabela verdade para o operador not:

Expressão	NOT Expressão
True	False
False	True



Combinando condições

Em situações em que precisamos verificar várias condições ao mesmo tempo, podemos fazer isso combinando condições com operadores lógicos. Veja:

Exemplo: Site de Vendas

Suponha que você esteja escrevendo um programa para um site de vendas on-line que dá desconto ao cliente se ele comprar mais de 5 itens **e** se o valor total da compra for superior a R\$ 200,00. Isso poderia ser representado pela seguinte condição:

if quantidade_de_itens > 5 and valor_total > 200: print("dar desconto")



Combinando condições

Em situações em que precisamos verificar várias condições ao mesmo tempo, podemos fazer isso combinando condições com operadores lógicos. Veja:

Exemplo: Aprovação em disciplina

Imagine um programa que precisa determinar se um aluno passou ou não em uma disciplina.

O aluno passa se tiver uma nota média maior que 7 **ou** se tiver uma nota maior que 5 **e** passar em um exame de recuperação:

```
if nota_media > 7 or (nota_media > 5 and passou_no_exame_de_recuperacao): print(Äluno passou")
```



Algoritmos



© Getty Images

Esse programa verifica se uma mensagem será enviada. A mensagem só será enviada **se** o usuário estiver conectado e **se** o destinatário for um amigo.

Leitura das entradas do usuário

```
conectado = input("O usuário está conectado (sim/não)? ").lower() amigo = input("O destinatário é um amigo (sim/não)? ").lower()
```

Verificação das condições e exibição da mensagem correspondente

```
if conectado == "sim" and amigo == "sim":
    print("Mensagem enviada")
else:
    print("Mensagem não enviada")
```



Vamos fazer uma atividade



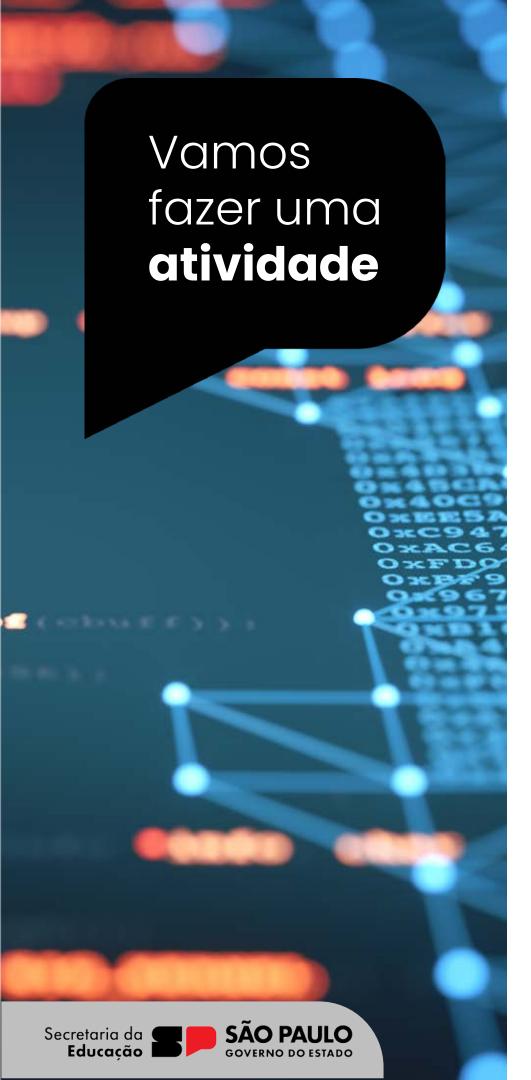


Análise de situaçãoproblema

O filme "O Jogo da Imitação" (2014) conta a história real de Alan Turing, um matemático britânico e criptógrafo que liderou uma equipe para decifrar o código Enigma, da Alemanha nazista, durante a Segunda Guerra Mundial.

- Desafio: imagine que você é parte da equipe de Turing e foi incumbido de aprimorar o processo de decifração do Enigma. A situação está crítica e cada minuto conta. Há uma quantidade imensa de mensagens sendo transmitidas todos os dias, e é humanamente impossível analisá-las manualmente.
- Qual é a solução? Desenvolver um algoritmo capaz de otimizar a decifração.





Exemplo – Como decifrar um enigma

6 Sexta letra do alfabeto → F

5 Quinta letra do alfabeto → E

50 Número 50 em algarismo romano → L

1 Número 1 em algarismo romano → I

26 Vigésima sexta letra do alfabeto → Z

MM Número MM (algarismo romano) em algarismo indo-arábico → 2000

R Posição da letra R no alfabeto → 18



Adaptado de WATTPAD SUSPENSELP, 2018. Elaborado especialmente para o curso.

Vamos fazer uma **atividade**

Essa atividade deverá ser publicada no local indicado pelo professor.



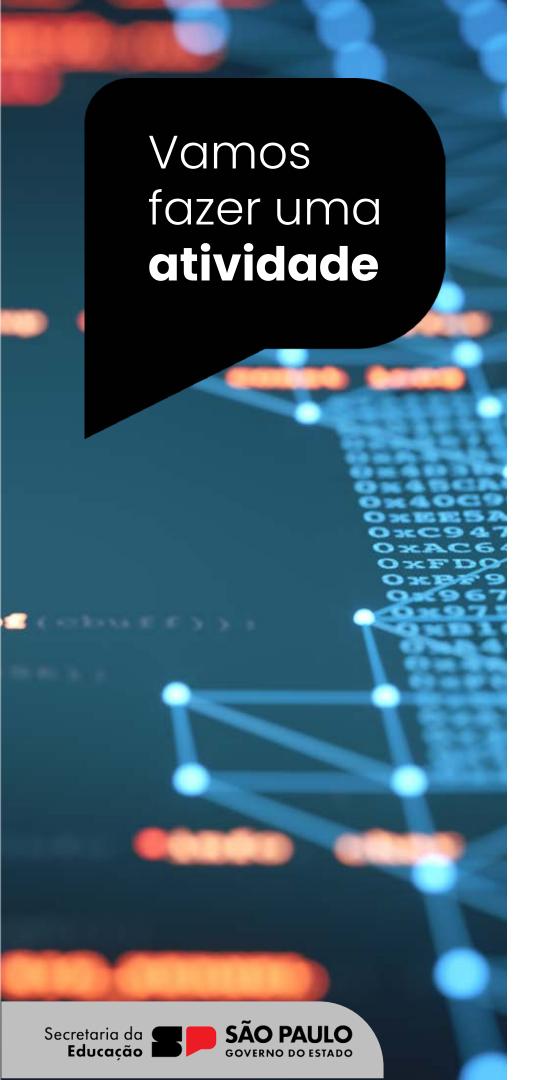


Em grupo de até 5 alunos

Questões norteadoras para solução do problema

- Quais seriam os principais elementos do algoritmo para decifrar os códigos de Enigma?
- 2 Como você usaria a lógica de programação para melhorar a eficiência do processo de decifração?
- O que deve ser considerado para garantir que o algoritmo seja o mais eficiente possível, considerando os limites de tempo e recursos computacionais da época?





Solução sugerida

Os principais elementos do algoritmo para decifrar os códigos de Enigma poderiam ser:

- estruturas de dados para armazenar o alfabeto e as configurações do Enigma;
 - funções para imitar a rotação dos rotores do Enigma;
 - uma série de operações para testar diferentes combinações de configurações do Enigma até encontrar a correta.
- 2 A lógica de programação poderia ser usada para melhorar a eficiência do processo de decifração.
- Seria importante considerar a otimização do algoritmo para reduzir o número de operações necessárias, eliminando, por exemplo, combinações de configurações que já foram testadas ou que são impossíveis.

Qual operador lógico você usaria para verificar se duas condições são verdadeiras ao mesmo tempo?

and

or

f

else



Qual operador lógico você usaria para verificar se duas condições são verdadeiras ao mesmo tempo?

and

else

RESPOSTA CORRETA!

O operador lógico "And" também pode ser substituído pela expressão "E", dessa forma, possibilita a verificação de duas situações simultâneas.



Qual é a estrutura de decisão que permite avaliar múltiplas condições sequencialmente?

if

while

for

else if



Qual é a estrutura de decisão que permite avaliar múltiplas condições sequencialmente?

if for

while

else if

RESPOSTA CORRETA!

Caso a primeira condição não contemple a necessidade do algoritmo, o Else If fica responsável por testar as demais condições.



No sistema de uma concessionária de automóveis, João deseja que o sistema mostre todos os carros que estejam entre os valores de 5 mil e 40 mil reais. Assinale a alternativa que melhor define essa situação.

valor >= 5000 e valor <= 40000

valor >= 5000 ou valor <= 40000

valor > 5000 e valor < 40000

valor > 5000 ou valor < 40000





No sistema de uma concessionária de automóveis, João deseja que o sistema mostre todos os carros que estejam entre os valores de 5 mil e 40 mil reais. Assinale a alternativa que melhor define essa situação.

valor >= 5000 e valor <= 40000

valor >= 5000 ou valor <= 40000

valor > 5000 e valor < 40000

valor > 5000 ou valor < 40000

Resposta correta

Está correta porque mostrará todos os carros que tenham valores entre 5000 e 40000 reais.





Hoje desenvolvemos:

- Entendimento sobre a estrutura de operadores lógicos e sua utilização em conjunto com as estruturas Condicionais;
- 2 Conhecimento sobre os principais operadores lógicos e sua função dentro do código;
- 3 A aplicação de algoritmos para situações cotidianas, permitindo o desenvolvimento de programas.



Referências da aula

PINHO, M. S. *Programação em Python*: Comandos de Decisão/Comandos de Seleção. Escola Politécnica – PUCRS. Disponível em: https://www.inf.pucrs.br/pinho/PCB/ComandosDeDecisao/Decisao.htm. Acesso em: 3 mar. 2024.

OTÁVIO MIRANDA. *Operadores Lógicos em Python* – Aula 4 (Parte 3). Disponível em: https://www.youtube.com/watch?v=MR2Q3I9qbaY. Acesso em: 3 mar. 2024.

WATTPAD SUSPENSELP. Desvendando Enigmas: Resposta Enigma 1, 1 jan. 2018. Disponível em: https://www.wattpad.com/516140216-desvendando-enigmas-resposta-enigma-1. Acesso em: 4 mar. 2024.

Identidade visual: Imagens © Getty Images



Educação Profissional Paulista

Técnico em
De sen volvim en to
de Sistem a s

