

Educação
Profissional
Paulista

Técnico em
Desenvolvimento
de Sistemas

Estruturas de repetição

Conceituação e laço de repetição: *while*

Aula 1

Código da aula: [SIS]ANO1C1B1S5A1

Exposição



Objetivo da aula

- Conhecer o conceito de estrutura de repetição com enfoque no método *while*.



Competências da Unidade (técnicas e socioemocionais)

- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento;
- Resolver problemas computacionais com estratégias ágeis.



Recursos didáticos

- Recurso audiovisual para a exibição de vídeos e imagens;
- Caderno para anotações.



Duração da aula

50 minutos.

Estruturas de repetição em programação: ênfase no laço *while*

As estruturas de repetição são fundamentais na programação para realizar tarefas repetidas sem a necessidade de reescrever o mesmo código várias vezes. O laço *while*, em particular, é uma ferramenta poderosa para repetir instruções até que uma condição especificada deixe de ser verdadeira, permitindo uma maior flexibilidade em comparação com estruturas de repetição fixas.

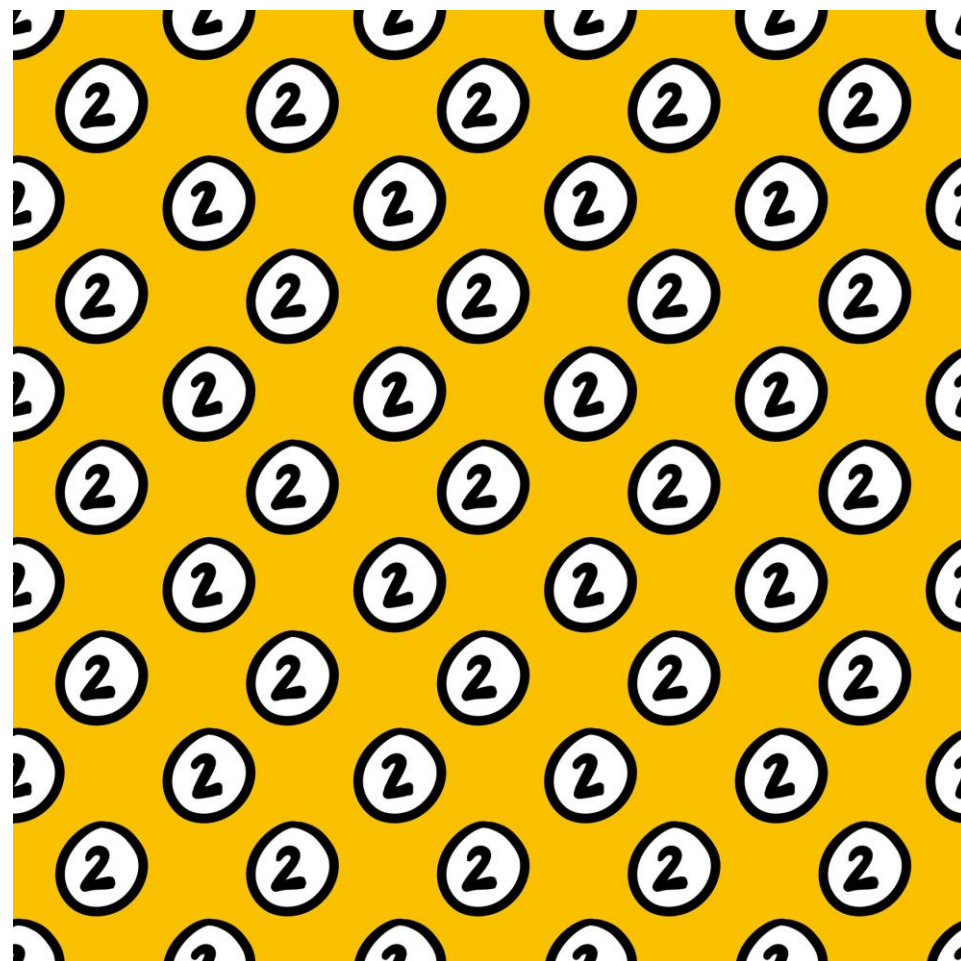
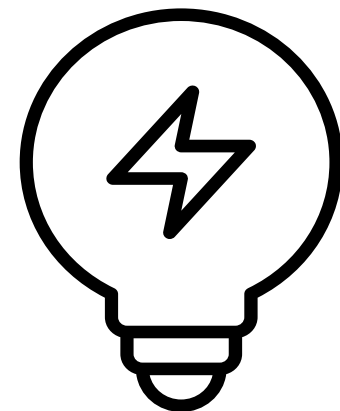
- ✓ Conceitos e aplicações gerais de estruturas de repetição
- ✓ Diferenças fundamentais entre *while*, *for* e *do-while*.
- ✓ Prevenção de laços infinitos e gestão de estados de *loop*.

Introdução à estrutura de repetição

Antes de definir o que são as estruturas de repetição, que tal fazermos um treino juntos?

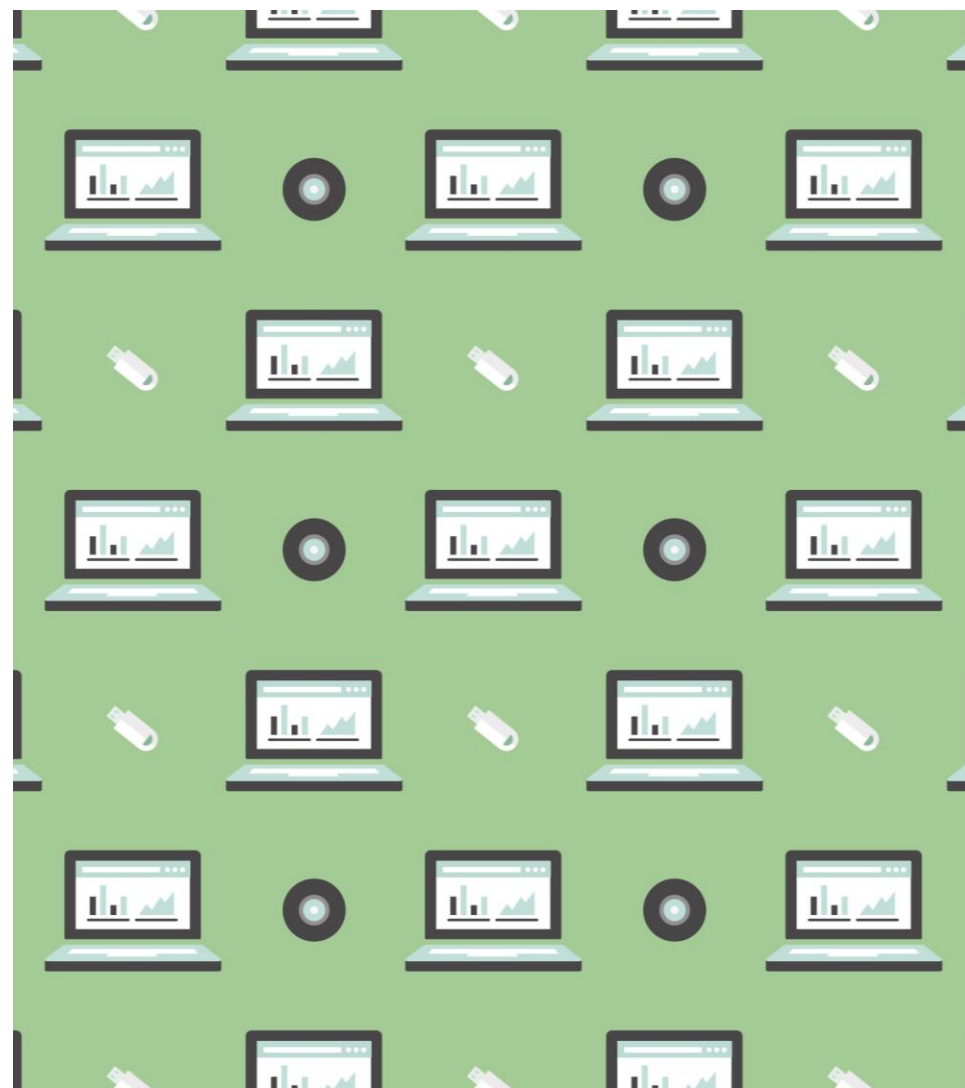
- Some os números de 2 em 2 manualmente até chegar em 100.

No início parece ser fácil, mas é cansativo e leva tempo.



© Getty Images

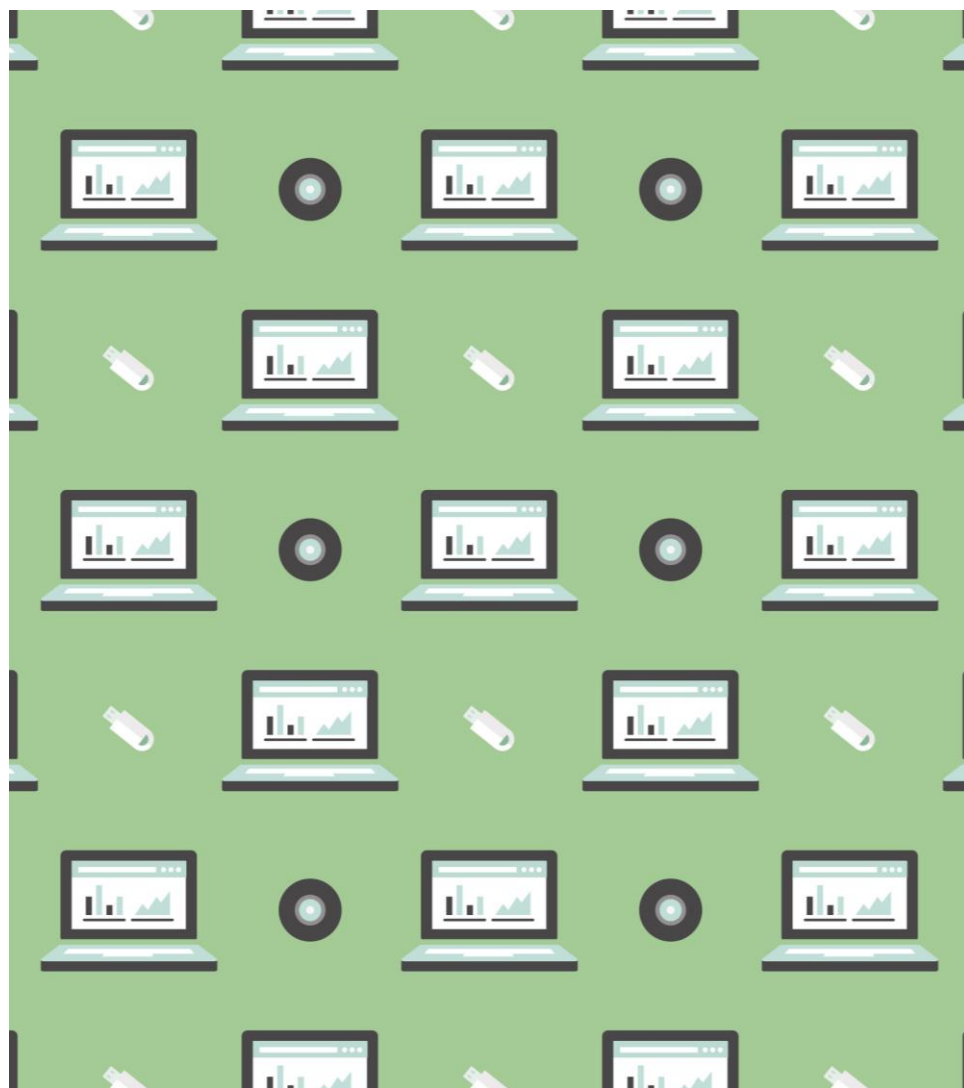
Introdução à estrutura de repetição



© Getty Images

Uma estrutura de repetição, também conhecida como laço ou *loop*, em programação, é uma construção que permite executar um bloco de código repetidamente, baseando-se em uma condição booleana: enquanto a condição for verdadeira, o bloco de código continua a executar. Quando a condição se torna falsa, o laço é encerrado e o fluxo de controle do programa se move para a próxima instrução após o laço.

Introdução à estrutura de repetição

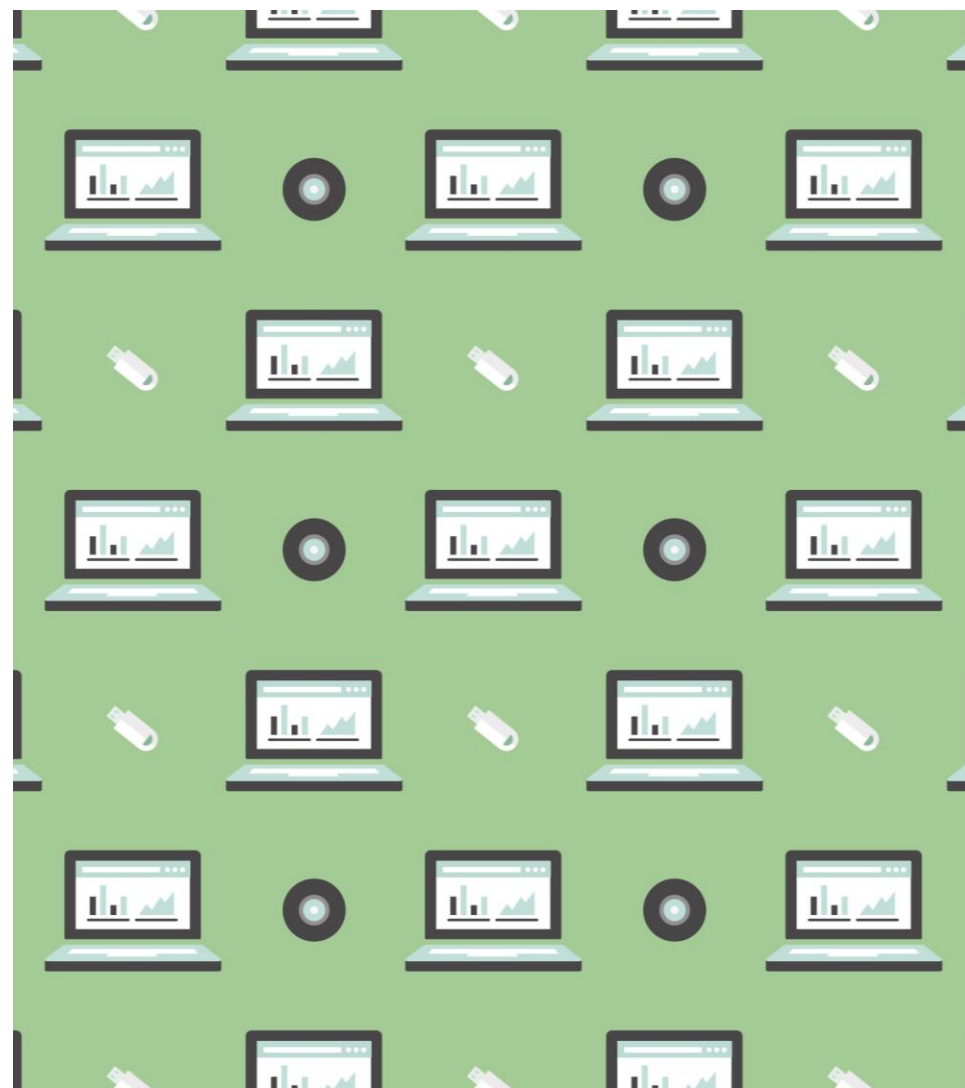


© Getty Images

Motivação para a criação de estruturas de repetição:

A motivação para se criarem estruturas de repetição surgiu da necessidade de eficiência e simplicidade no código. Antes da existência de laços, se quiséssemos executar a mesma ação várias vezes, precisaríamos escrever a mesma linha de código tantas vezes quanto fosse necessário. Além de ser uma prática propensa a erros, isso tornava os programas longos, difíceis de se ler e de se manter.

Introdução à estrutura de repetição



© Getty Images

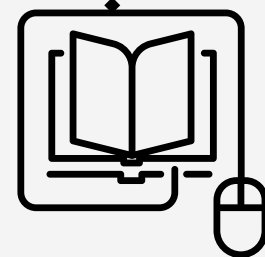
Por exemplo, suponhamos que você queira imprimir os números de 1 a 10.

Sem um laço, você teria que escrever dez linhas de comandos de impressão. Com um laço, você pode reduzir isso para apenas algumas linhas, independentemente de quantas vezes deseja imprimir o número, tornando o código mais conciso e fácil de alterar.

Tipos de estruturas de repetição

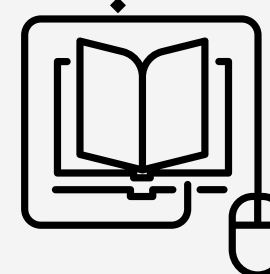
Laços fixos (*for*):

São usados quando se sabe previamente quantas vezes o código deve ser executado. A repetição é baseada em um contador ou em uma coleção de elementos.



Laços condicionais (*while* e *do-while*):

São utilizados quando o número de repetições depende de condições que podem não ser conhecidas até que o código seja executado. Eles são muito úteis para situações como leitura de dados até que um determinado valor seja encontrado, ou processamento contínuo até que uma condição externa seja satisfeita.



Exposição

Onde esse conceito pode ser encontrado?



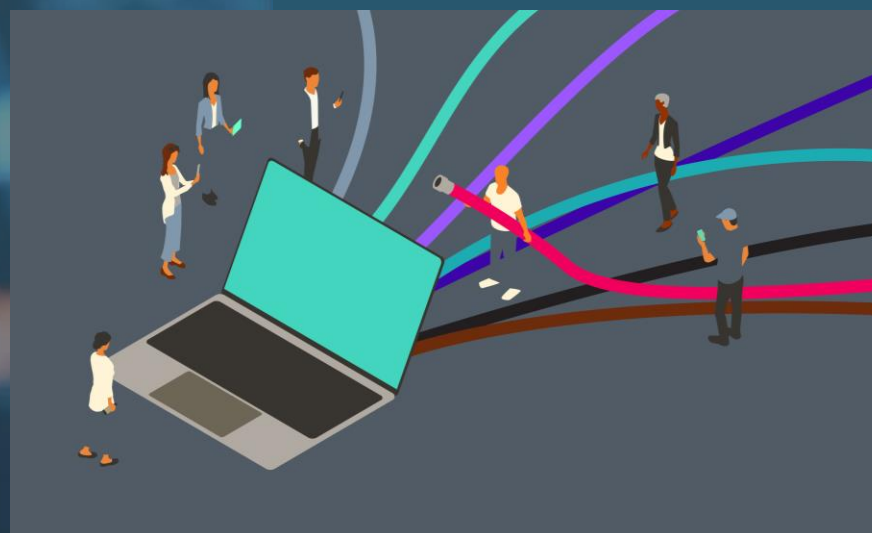
Processamento de entradas de dados

Ler dados de um usuário ou arquivo até que seja encontrado um terminador específico ou até que os dados se esgotem.



Jogos

Executar o *loop* de jogo, que continua renderizando cada *frame* e processando entrada de usuário até que o jogo termine.



Serviços de rede

Manter servidores escutando por conexões de entrada indefinidamente até que sejam desligados.

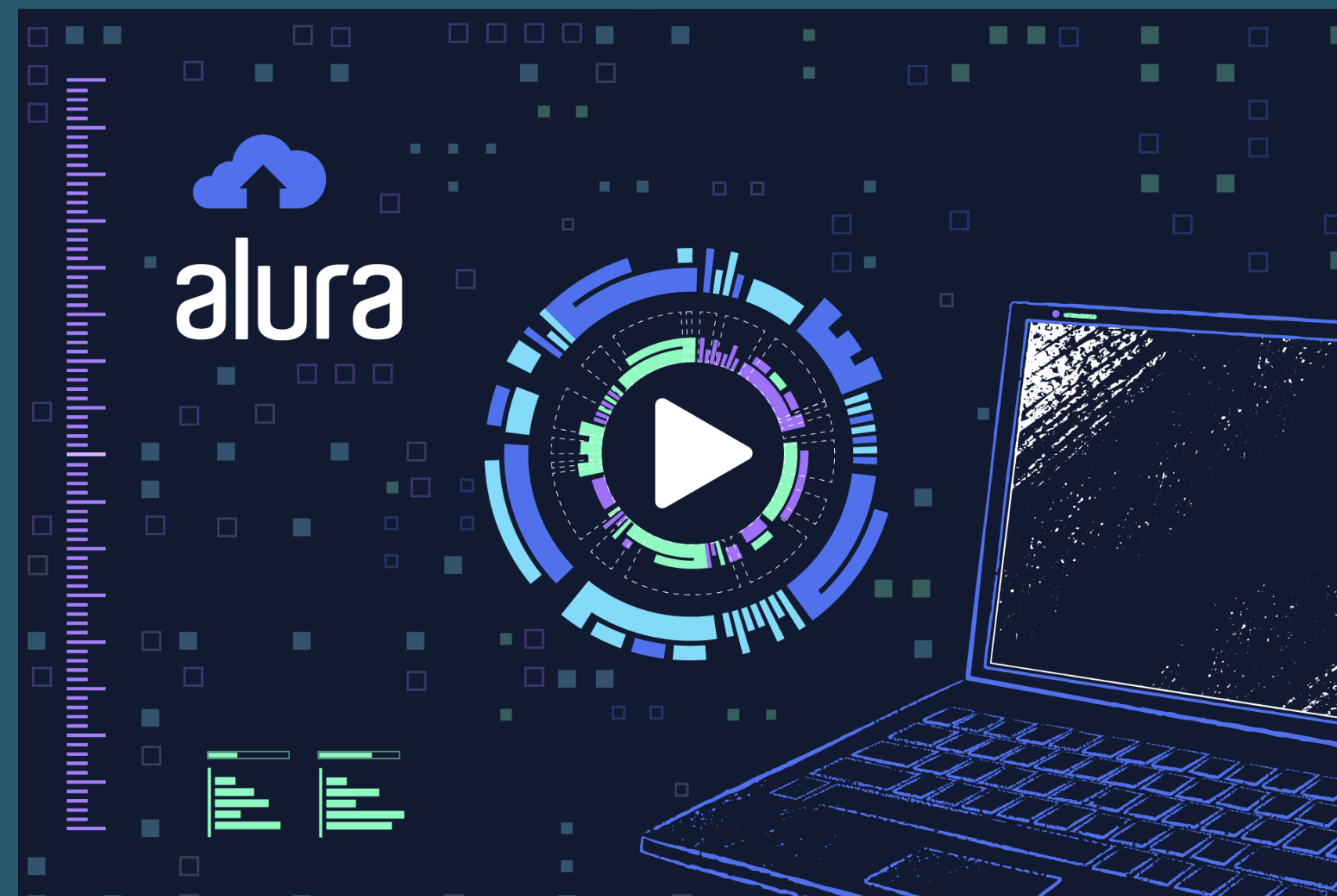


© Getty Images

Exposição



Estrutura de repetição na prática!



Acesso para link externo:
ALURA. JavaScript e HTML: desenvolva um jogo e pratique lógica de programação: 01 Repetir enquanto...
Disponível em: <https://cursos.alura.com.br/course/logica-programacao-javascript-html/task/17737>.
Acesso em: 26 jan. 2024. Acesso para link externo.

Tipos de estruturas de repetição

Em programação, há três tipos principais de laços de repetição que são comumente utilizados em diversas linguagens de programação: *for*, *while* e *do-while*.

Laço *for*: usado quando o número de iterações é conhecido antes do início do laço. É ideal para iterar sobre coleções de dados ou sequências em que o tamanho é predefinido.



Exemplo em Python

```
for i in range(5): # Itera de 0 a 4  
    print(i)
```

Tipos de estruturas de repetição

Em programação, há três tipos principais de laços de repetição que são comumente utilizados em diversas linguagens de programação: *for*, *while* e *do-while*.

Laço *while*: apropriado quando o número de iterações não é conhecido e a decisão de sair do laço depende de condições que são avaliadas durante a execução do laço.



Exemplo em Python

```
i = 0
while i < 5: # Executa enquanto i for menor que 5
    print(i)
    i += 1
```

Tipos de estruturas de repetição

Em programação, há três tipos principais de laços de repetição que são comumente utilizados em diversas linguagens de programação: *for*, *while* e *do-while*.

Laço *do-while*: similar ao *while*, mas garante que o corpo do laço seja executado pelo menos uma vez, pois a condição é verificada após a execução do corpo do laço. A linguagem Python não tem um laço *do-while* nativo, mas outras linguagens, como C, C++, e Java, têm.



Exemplo em C

```
int i = 0;
do {
    cout << i << endl;
    i++;
} while (i < 5);
```


Vamos
fazer um
quiz



1. Um desenvolvedor precisa escrever um programa que continuará executando enquanto receber dados de sensores de temperatura que não sigam um intervalo fixo de envio. O programa deve processar os dados assim que os receber e só terminar quando um sinal de "desligamento" for recebido. Qual estrutura de repetição é mais apropriada para implementar esse requisito?

Vamos
fazer um
quiz



Laço *for*, devido à sua capacidade de executar um número predeterminado de vezes.

Laço *while*, porque ele pode executar baseado na recepção de um sinal de condição variável.

Laço *do-while*, para garantir que os dados sejam processados pelo menos uma vez antes de verificar o sinal de desligamento.

Nenhuma estrutura de repetição; o uso de interrupções (*interrupts*) é o mais indicado para lidar com a recepção de dados.

Vamos
fazer um
quiz



Laço *for*, devido à sua capacidade de executar um número predeterminado de vezes.

Laço *while*, porque ele pode executar baseado na recepção de um sinal de condição variável.

Laço *do-while*, para garantir que os dados sejam processados pelo menos uma vez antes de verificar o sinal de desligamento.

Nenhuma estrutura de repetição; o uso de interrupções (*interrupts*) é o mais indicado para lidar com a recepção de dados.



Vamos
fazer um
quiz



2. Suponha que um sistema de atendimento ao cliente automatizado deve operar continuamente, mas permitir a intervenção de um operador humano quando uma situação que requer atenção especial é detectada. Qual estrutura de repetição é mais apropriada para manter o sistema em execução contínua até que a intervenção seja necessária?

Vamos
fazer um
quiz

Laço *for*, já que a intervenção pode ser programada para ocorrer após um certo número de atendimentos.

Laço *while*, porque pode continuar indefinidamente e checar constantemente por situações que requerem atenção.

Laço *do-while*, assegurando que cada cliente receba atendimento antes de qualquer verificação.

Laço *for-in*, para iterar sobre todos os clientes de uma lista de espera predefinida.



Vamos
fazer um
quiz



Laço *for*, já que a intervenção pode ser programada para ocorrer após um certo número de atendimentos.

Laço *while*, porque pode continuar indefinidamente e checar constantemente por situações que requerem atenção.

Laço *do-while*, assegurando que cada cliente receba atendimento antes de qualquer verificação.

Laço *for-in*, para iterar sobre todos os clientes de uma lista de espera predefinida.



Vamos
fazer um
quiz



3. Durante o desenvolvimento de um software de processamento de transações financeiras, os desenvolvedores se deparam com a necessidade de processar uma fila de transações em que novas transações podem ser adicionadas a qualquer momento. Qual estrutura de repetição permite que o software continue a processar a fila enquanto novas transações estão chegando?

Vamos
fazer um
quiz



Laço *for*, pois permite a iteração até que a fila esteja vazia.

Laço *while*, devido à sua habilidade de se adaptar a uma fila em constante mudança.

Laço *do-while*, garantindo que cada transação seja processada pelo menos uma vez.

Estruturas de repetição são inadequadas; eventos assíncronos são a solução ideal para processamento em tempo real.

Vamos
fazer um
quiz



Laço *for*, pois permite a iteração até que a fila esteja vazia.

Laço *while*, devido à sua habilidade de se adaptar a uma fila em constante mudança.

Laço *do-while*, garantindo que cada transação seja processada pelo menos uma vez.

Estruturas de repetição são inadequadas; eventos assíncronos são a solução ideal para processamento em tempo real.



Hoje desenvolvemos:

- 1 As principais estruturas de repetição e suas funções principais dentro do fluxo de programação.
- 2 A possibilidade de utilização do laço de repetição *while* e suas características.
- 3 O formato de aplicação da estrutura de repetição em um projeto real voltado para páginas da web.

O que nós
**aprendemos
hoje?**

© Getty Images



Saiba mais

Que tal aprofundar-se um pouco mais em estruturas de repetição utilizando a linguagem Python, que será amplamente abordada dentro de nossos projetos?

DANTAS, R. Python básico: estruturas de repetição. **Medium**, 30 nov. 2020. Disponível em: <https://medium.com/renato-dantas/python-b%C3%AAsico-estruturas-de-repeti%C3%A7%C3%A3o-bd9c2cc303c3>. Acesso em: 26 jan. 2024.

Referências da aula

DANTAS, R. **Python básico:** estruturas de repetição. **Medium**, 30 nov. 2020. Disponível em: <https://medium.com/renato-dantas/python-b%C3%AAsico-estruturas-de-repeti%C3%A7%C3%A3o-bd9c2cc303c3>. Acesso em: 26 jan. 2024.

LABENU. **Aula 08:** Estruturas de Repetição – Lógica de Programação para Iniciantes. Disponível em: <https://www.youtube.com/watch?v=gUGQ4FYleQo>. Acesso em: 26 jan. 2024.

Identidade visual: imagens © Getty Images

Educação
Profissional
Paulista

Técnico em
Desenvolvimento
de Sistemas