

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Estruturas de Repetição

Atividades práticas – Criação de programas

Aula 4

Código da aula: [SIS]ANO1C1B2S10A4

Exposição



Objetivos da aula

- Compreender as aplicações do laço *while* em diferentes linguagens de programação.



Competências da unidade (técnicas e socioemocionais)

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



Recursos didáticos

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



Duração da aula

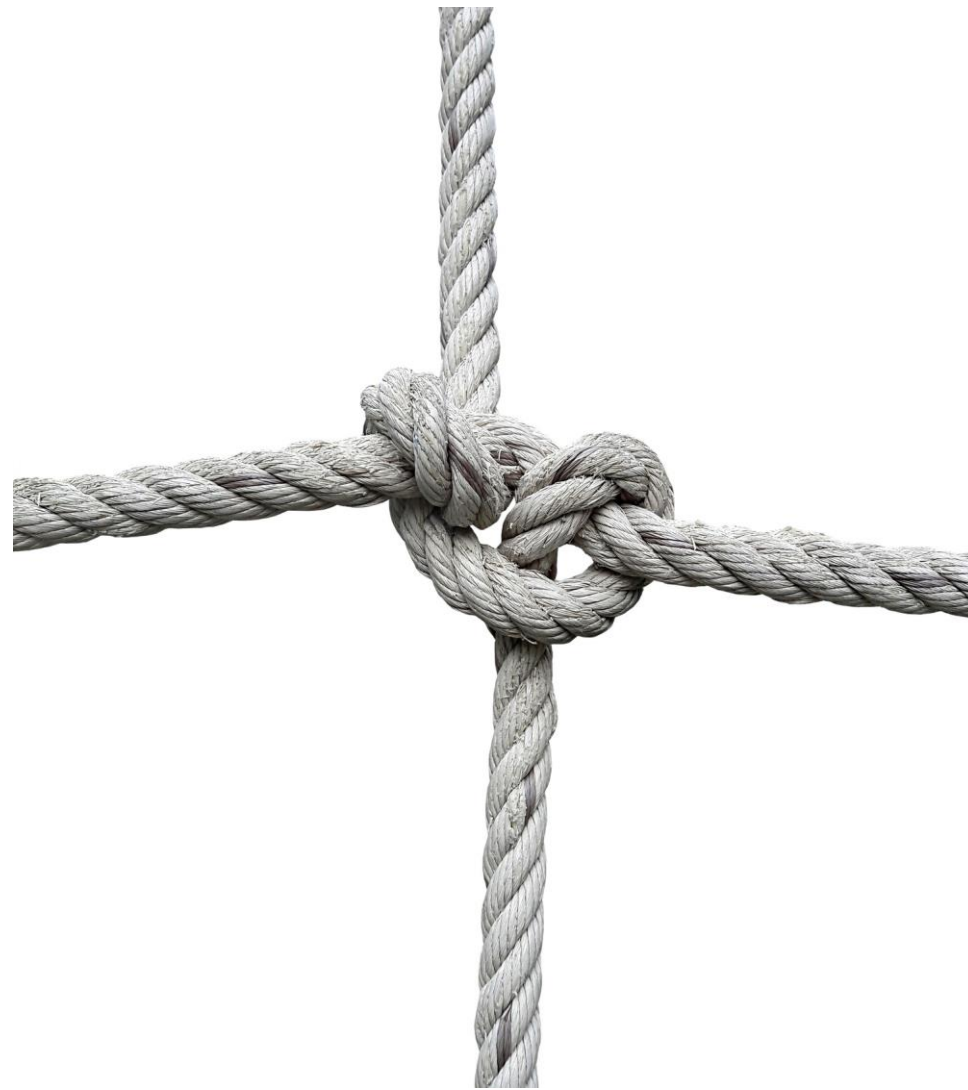
50 minutos.

Integração de laços de repetição com estruturas de decisão

A habilidade de integrar laços de repetição com estruturas de decisão é fundamental para a criação de programas complexos e reativos. Isso permite que os programadores escrevam códigos que podem se adaptar a diferentes condições e eventos enquanto operam dentro de ciclos de repetição. A compreensão dessa relação é particularmente vital para resolver problemas que requerem lógica condicional dentro de repetições para decisões dinâmicas.

- ✓ Aplicação prática de estruturas de decisão;
- ✓ Uso conjunto com laço *while*;
- ✓ Prevenção e manejo de laços infinitos com decisões.

Exposição



© Getty Images

Vamos lembrar a definição e a relação entre laços e decisões

Laços e decisões são dois conceitos fundamentais na programação que, quando combinados, permitem a criação de programas dinâmicos e interativos. Laços, ou loops, são usados para repetir um conjunto de instruções até que uma condição específica seja atendida, enquanto estruturas de decisão direcionam o fluxo de execução com base no resultado de uma condição booleana.

Laços de repetição como o ***while*** e o ***for*** dependem fortemente de avaliações condicionais para determinar a continuidade da repetição. No laço *while*, a condição é verificada antes de cada iteração, e o loop continua até que a condição avalie como falsa. No caso do laço *for*, a condição é geralmente uma variável contadora que atinge um valor limite.

Exposição

Boas práticas na combinação de laços e decisões



© Getty Images

Ao combinar laços e estruturas de decisão, as seguintes práticas são recomendadas para manter a clareza e a eficiência do código:

Minimize a lógica dentro de laços: mantenha o corpo do laço tão simples quanto possível. Se a lógica de decisão se torna complexa, considere mover parte dela para uma função separada.

Use break e continue com cuidado: embora sejam úteis para controlar o fluxo de execução, o uso excessivo dessas instruções pode tornar o código difícil de seguir e manter.

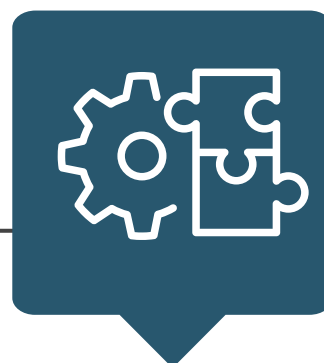
Evite modificar o contador ou a condição do laço externamente: isso pode levar a bugs sutis e difíceis de depurar. Qualquer variável que controle o laço deve ser alterada de maneira previsível e transparente.

Vamos
fazer uma
atividade

Vamos trabalhar dois
exercícios práticos
sobre repetição.

 **35 minutos**

Exercícios práticos – Estrutura de repetição



Materiais necessários

Os materiais necessários para a
realização da atividade são:

- Computador com acesso à internet;
- Caderno ou folhas de sulfite para anotações e rascunhos;
- Lápis ou canetas para anotações.



Passo a passo

Para executar a atividade prática, siga
estes passos:

1. Abra o arquivo complementar com a explicação da atividade;
2. Leia com atenção o enunciado de cada exercício, anotando os pontos principais;
3. Crie e execute os códigos propostos nas ferramentas de desenvolvimento em Python.



Faça download da explicação da atividade

Hoje desenvolvemos:

- 1 A prática do relacionamento entre fluxos de decisão e fluxos de repetição de códigos.
- 2 A decisão, por meio de um exemplo prático, da utilização do laço *while* ou *for* de acordo com o contexto necessário.
- 3 A prática da refatoração do código a partir de um erro comum que pode ocorrer ao se trabalhar com estruturas de repetição.

O que nós
**aprendemos
hoje?**

© Getty Images



Saiba mais

Para avançar um pouco mais no conceito de estruturas de repetição, conheça também o conceito de *format*, para aprimorar seus códigos.

HASHTAG PROGRAMAÇÃO. *Estruturas de repetição e o format* - Curso de Python – Aula 3. Disponível em:

<https://www.youtube.com/watch?v=ft8OJam4VOw>.

Acesso em: 25 fev. 2024.

```
void _decode_(char cbuff **buff)
{
    if (step == AES_LOC_PASS) {
        src = cbuff->load();
        dest = getattr(&ptr, &mod,
            if (mod != NULL) as dest)
        dest += buffer->TABLE(mod)
        mask |= (previous_aes12
            if (mask & SIG_KERNEL) !
        return _ERROR_
        return NULL
    }
}
```

Referências da aula

CURSO EM VÍDEO. *Curso Python #014 – Estrutura de repetição while*.

Disponível em: <https://www.youtube.com/watch?v=LH6Oln2lBaI>. Acesso em: 25 fev. 2024.

HASHTAG PROGRAMAÇÃO. *Estruturas de repetição e o format – Curso de Python – Aula 3*. Disponível em:

<https://www.youtube.com/watch?v=ft8OJam4VOw>. Acesso em: 25 fev. 2024.

LOPES, E. *Loops e estruturas de repetição no Python*. Python Academy, 2021.

Disponível em: <https://pythonacademy.com.br/blog/estruturas-de-repeticao>. Acesso em: 25 fev. 2024.

Identidade visual: © Getty Images.

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**