

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**

# Lógica de programação e algoritmos

## Estruturas de controle: repetição

### Aula 1

**Código da aula: [DADOS]ANO1C3B2S12A1**

# Exposição



## Objetivo da Aula

Explorar e apresentar, com apoio de pseudocódigos, os conceitos importantes de estruturas de controle de repetição.



## Competências da Unidade (Técnicas e Socioemocionais)

- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências;
- Compreender e dominar técnicas de manipulação de dados; extrair, transformar e carregar conjuntos de dados de diferentes fontes, garantindo a qualidade e a integridade dos dados; criar e compreender visualizações gráficas.



## Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens.



## Duração da Aula

50 minutos

# Exposição

## Recapitulação de algoritmos e lógica de programação

```
if tar_word in list_word and seed_word_p in list_wo
    count_word = len(list_word)

if count_word <= NEAR: # if number of word in e
    count_p += 1
else:
    positive_tar = []
    positive_sw = []
    for l_index, l_word in enumerate(list_word)
        if tar_word == l_word:
            positive_tar.append(l_index)
        elif seed_word_p == l_word:
            positive_sw.append(l_index)

    breaker = False
    for index_tar in positive_tar:
```

© Getty Images

- ✓ **Definição:** algoritmos são conjuntos de passos organizados, semelhantes a receitas, que guiam a resolução de problemas de forma eficiente;
- ✓ **Uso:** fundamentais na computação, algoritmos são "mapas" lógicos que possibilitam desde cálculos simples até inovações tecnológicas avançadas, como reconhecimento de voz;
- ✓ **Importância:** são a base de todos os softwares, permitindo a automação de tarefas e o desenvolvimento de soluções complexas.



## Exposição

# Recapitulação de algoritmos e lógica de programação

## A importância de fluxogramas e de pseudocódigos

Eles otimizam o desenvolvimento, proporcionando uma compreensão visual e lógica do processo, que facilita a identificação de melhorias e acelera a transição para a implementação prática, independentemente da linguagem de programação ou do software usado para implementação.



© Getty Images


# Recapitulação de estrutura sequencial

O que é uma estrutura sequencial?

- ✓ É a forma mais simples de **controle de fluxo em programação**;
- ✓ Consiste em uma **sequência linear de instruções**, em que uma ação ocorre após a outra, de maneira ordenada;
- ✓ É fundamental para executar **tarefas passo a passo**, uma após a outra, sem desvios controlados.

## Recapitulação de pseudocódigos

### Exemplo visto anteriormente (troca de lâmpada)

- 
- pegar uma escada
  - posicionar a escada embaixo da lâmpada
    - buscar uma lâmpada reserva
    - subir na escada
    - retirar a lâmpada queimada
    - inserir a lâmpada reserva

Estrutura  
sequencial dos  
pseudocódigos

Elaborado especialmente para o curso.



### Tome nota

Os pseudocódigos permitem que a visualização das estruturas sequenciais sejam compreendidas exatamente como será no programa real. Elas seguem linha a linha, assim como nós fazemos leitura, da esquerda para a direita, de cima para baixo.

## Estruturas de repetição

### O que são?

- ✓ São conhecidas também como **laços** ou **loops**,
- ✓ São recursos fundamentais em programação, que permitem a execução repetida de um bloco de código enquanto uma condição específica for atendida.

Existem duas principais estruturas de repetição:

**Laço "Enquanto" (While):** repete um bloco de código enquanto uma condição é verdadeira.

**Laço "Para" (For):** executa um bloco de código em um número específico de vezes.



## Estruturas de repetição

Importância das estruturas de repetição:



### **Eficiência**

Permitem a automação de tarefas repetitivas, economizando tempo e esforço no desenvolvimento de software;



### **Flexibilidade**

Adaptam-se a situações em que a quantidade de iterações não é conhecida antecipadamente, garantindo que o código execute o número adequado de vezes;



### **Manutenção**

Simplificam a manutenção do código, pois reduzem a duplicação de instruções e tornam as atualizações mais consistentes;



### **Solução para problemas específicos**

São essenciais para resolver problemas que exigem processamento repetitivo, como processamento de listas, cálculos iterativos e interação com conjuntos de dados;



### **Redução de erros**

Ao evitar a repetição manual de código, as estruturas de repetição diminuem a probabilidade de erros.

# Estruturas de repetição – ENQUANTO/FAÇA

O que é?

A **estrutura de repetição "Enquanto" (While)** é uma construção fundamental em programação que permite a execução de um bloco de código **enquanto uma condição específica permanece verdadeira**.



## Na prática

- O bloco de código dentro do "Enquanto" é executado repetidamente enquanto a condição especificada permanecer verdadeira;
- A condição é avaliada antes da execução do bloco e, se for falsa inicialmente, o bloco não será executado.

## Estruturas de repetição – ENQUANTO/FAÇA

Como utilizar?

ENQUANTO (condição) FAÇA

// Bloco de código a ser repetido

FIM ENQUANTO



### Atenção!

**Evite:** é importante que essa condição seja sempre trabalhada dentro da estrutura de repetição, caso contrário ela estará sempre em execução e isso é um problema grave. Portanto, atente aos laços que possam ser infinitos.



# Estruturas de repetição – ENQUANTO/FAÇA

## Enunciado 1:

1. Escreva um programa em pseudocódigo que utilize a estrutura de repetição "Enquanto" para realizar a contagem de 1 a 5.
2. Crie um pseudocódigo para essa situação.

## Requisitos:

- necessário iniciar a variável usada no "Enquanto" com valor antes da estrutura;
- alterar a variável da condição de repetição para evitar laços infinitos.



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – ENQUANTO/FAÇA

Solução em pseudocódigo:

1. contador  $\leftarrow$  1
2. ENQUANTO contador  $\leq$  5 FAÇA
3.   Escrever "Contagem: ", contador
4.   contador  $\leftarrow$  contador + 1
5. FIM ENQUANTO



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – ENQUANTO/FAÇA

## Enunciado 2:

1. Escreva um programa em pseudocódigo que utilize a estrutura de repetição "Enquanto" para realizar uma contagem regressiva de 5 a 1.
2. Crie um pseudocódigo para essa situação.

## Requisitos:

- necessário iniciar a variável usada no “Enquanto” com valor antes da estrutura;
- alterar a variável da condição de repetição para evitar laços infinitos.



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – ENQUANTO/FAÇA

Solução em pseudocódigo:

1. contador  $\leftarrow$  5
2. ENQUANTO contador  $\geq$  1 FAÇA
3.   Escrever "Contagem: ", contador
4.   contador  $\leftarrow$  contador - 1
5. FIM ENQUANTO



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – ENQUANTO/FAÇA

## Enunciado 3:

1. Crie um programa em pseudocódigo que utilize a estrutura de repetição "Enquanto" para solicitar ao usuário a entrada de uma senha.
2. Enquanto a senha digitada não for igual a "1234", o programa deve continuar pedindo a senha. Quando a senha correta for inserida, exiba a mensagem "Acesso concedido".

## Requisitos:

- necessário iniciar a variável senhaDigitada e a constante senhaCorreta.



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – ENQUANTO/FAÇA

Solução em pseudocódigo:

1. senhaCorreta <- "1234"
2. senhaDigitada <- ""
3. ENQUANTO senhaDigitada <> senhaCorreta FAÇA
4.   Escrever "Digite a senha:"
5.   Ler senhaDigitada
6. FIM ENQUANTO
7. Escrever "Acesso concedido"



Momento  
de **reflexão**

© Pexels



Vamos  
fazer um  
**quiz**

**Qual é o principal objetivo das estruturas de repetição em programação?**

**Realizar operações aritméticas**

**Controlar o fluxo de execução de um programa**

**Declarar variáveis e constantes**

**Realizar operações lógicas**





Vamos  
fazer um  
**quiz**

## Qual é o principal objetivo das estruturas de repetição em programação?

Realizar operações aritméticas

Controlar o fluxo de execução de um programa ✓

Declarar variáveis e constantes

Realizar operações lógicas

### RESPOSTA CORRETA!

Destaca o principal propósito das estruturas de repetição, que é controlar a repetição de um bloco de código com base em uma condição.



Vamos  
fazer um  
**quiz**

**Ao utilizar a estrutura de repetição "Enquanto" (While) em um algoritmo, qual é uma prática que deve ser evitada para garantir a correta execução do loop?**

**Iniciar a variável de controle fora da condicional do "Enquanto"**

**Modificar a variável de controle dentro da condição do "Enquanto"**

**Utilizar critérios de saída complexos dentro da condicional do "Enquanto"**

**Declarar novas variáveis dentro da condicional do "Enquanto"**





Vamos  
fazer um  
**quiz**

**Ao utilizar a estrutura de repetição "Enquanto" (While) em um algoritmo, qual é uma prática que deve ser evitada para garantir a correta execução do loop?**

**Iniciar a variável de controle fora da condicional do "Enquanto"**

**Modificar a variável de controle dentro da condição do "Enquanto"**



**Utilizar critérios de saída complexos dentro da condicional do "Enquanto"**

**Declarar novas variáveis dentro da condicional do "Enquanto"**

### **RESPOSTA CORRETA!**

Essa alternativa é a correta porque modificar a variável de controle dentro da condição do "Enquanto" pode levar a resultados inesperados e, potencialmente, causar loops infinitos.



Vamos  
fazer um  
**quiz**

**Considere o seguinte algoritmo em pseudocódigo:**

```
INÍCIO  
  soma ← 0  
  contador ← 1  
  ENQUANTO contador ≤ 5 FAÇA  
    soma ← soma + contador  
    contador ← contador + 1  
  FIM ENQUANTO  
  Escrever "Soma: ", soma  
FIM
```

**Soma: 15**

**Soma: 10**

**Soma: 5**

**Soma: 20**





Vamos  
fazer um  
**quiz**

## Considere o seguinte algoritmo em pseudocódigo:

```
INÍCIO  
  soma ← 0  
  contador ← 1  
  ENQUANTO contador ≤ 5 FAÇA  
    soma ← soma + contador  
    contador ← contador + 1  
  FIM ENQUANTO  
  Escrever "Soma: ", soma  
FIM
```



**Soma: 15**

**Soma: 10**

**Soma: 5**

**Soma: 20**

### RESPOSTA CORRETA!

O algoritmo realiza a soma dos números de 1 a 5, resultando em 15. Enquanto as opções B, C e D representam valores incorretos da soma, considerando o algoritmo apresentado.



O que nós  
**aprendemos  
hoje?**

© Getty Images

## Hoje desenvolvemos:

- 1** Conhecimento que a estrutura de repetição "Enquanto" (While) é uma construção fundamental em programação que permite a execução de um bloco de código enquanto uma condição específica permanece verdadeira;
- 2** Compreensão de que é importante que essa condição seja sempre trabalhada dentro da estrutura de repetição, caso contrário ela estará sempre em execução e isso é um problema grave. Portanto, atente aos laços que possam ser infinitos;
- 3** Prática com QUIZ e com exercícios sem avaliação; solução de algoritmos com Estruturas de Controle de Repetição "Enquanto", que são largamente utilizadas na indústria para a construção de algoritmos profissionais.





# Saiba mais

Entendendo o básico de Python para treinar estruturas de Controle – Parte 1

ALURA. *Python para Data Science: primeiros passos.*

01. Começando com Python. Disponível em:

<https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122382>. Acesso em:

15 mar. 2024.

Entendendo o básico de Python para treinar estruturas de Controle – Parte 2

ALURA. *Python para Data Science: primeiros passos.*

02. Manipulando dados no Python. Disponível em:

<https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/123730>. Acesso em:

15 mar. 2024.

## Referências da aula

ALURA. *Python para Data Science: primeiros passos*. 01. Começando com Python. Disponível em: <https://cursos.alura.com.br/course/python-data-science-primarios-passos/task/122382>. Acesso em: 15 mar. 2024.

ALURA. *Python para Data Science: primeiros passos*. 02. Manipulando dados no Python. Disponível em: <https://cursos.alura.com.br/course/python-data-science-primarios-passos/task/123730>. Acesso em: 15 mar. 2024.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. *Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python*. Porto Alegre: Bookman, 2022.

Identidade visual: Imagens © Getty Images



# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**



## S12 – Aula 1 – Quiz

Condições de conclusão

Ver

Qual é o principal objetivo das estruturas de repetição em programação?

- ☐ Controlar o fluxo de execução de um programa
- ☐ Realizar operações lógicas
- ☐ Declarar variáveis e constantes
- ☐ Realizar operações aritméticas

Ao utilizar a estrutura de repetição "Enquanto" (While) em um algoritmo, qual é uma prática que deve ser e execução do loop?

- ☐ Declarar novas variáveis dentro da condicional do "Enquanto"
- ☐ Modificar a variável de controle dentro da condição do "Enquanto"
- ☐ Utilizar critérios de saída complexos dentro da condicional do "Enquanto"
- ☐ Iniciar a variável de controle fora da condicional do "Enquanto"

Considere o seguinte algoritmo em pseudocódigo:

INÍCIO

soma <- 0

contador <- 1

ENQUANTO contador <= 5 FAÇA

soma <- soma + contador

contador <- contador + 1

FIM ENQUANTO

Escrever "Soma: ", soma

FIM

- ☐ Soma: 5
- ☐ Soma: 10
- ☐ Soma: 20
- ☐ Soma: 15



**Disciplina**

Lógica, Algoritmo e Operação de Planilhas Eletrônicas 2º Bimestre

**Curso**

Técnico em Ciência de Dados

**Ano letivo**

2025



[Retornar ao Sumário](#)

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**



# **Lógica de programação e algoritmos**

## **Estruturas de controle: repetição**

### **Aula 2**

**Código da aula: [DADOS]ANO1C3B2S12A2**

# Exposição



## Objetivo da Aula

Praticar o uso de estruturas de controle de repetição com exercícios, com e sem avaliação pelo AVA.



## Competências da Unidade (Técnicas e Socioemocionais)

- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências;
- Compreender e dominar técnicas de manipulação de dados; extrair, transformar e carregar conjuntos de dados de diferentes fontes, garantindo a qualidade e a integridade dos dados; criar e compreender visualizações gráficas.



## Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens.



## Duração da Aula

50 minutos



# Estruturas de repetição – PARA/FAÇA

## Funcionamento

- A variável de controle é inicializada com um valor específico;
- O bloco de código dentro do "Para" é executado repetidamente enquanto a variável de controle estiver dentro do intervalo especificado;
- A variável de controle é incrementada ou decrementada em cada iteração, de acordo com o valor do "PASSO".

# Estruturas de repetição – PARA/FAÇA

O que é?

A **estrutura de repetição "Para"** é uma construção fundamental em programação que facilita a iteração sobre um conjunto de valores.

Note que há mais de uma forma de utilizar essa estrutura de repetição:

- ✓ Uma **interage item a item** de uma lista de itens;
- ✓ Outra **interage com números** de forma pré-determinada.

## Estruturas de repetição – PARA/FAÇA

Como utilizar?

**PARA** variável **DE** valor\_inicial **ATÉ** valor\_final **PASSO** incremento **FAÇA**  
// Bloco de código a ser repetido  
**FIM PARA**



### Atenção!

É necessário garantir que a variável de controle seja atualizada corretamente no final de cada iteração para evitar loops infinitos.



# Estruturas de repetição – PARA/FAÇA

### Importância:

- a estrutura "Para" é particularmente útil quando o número exato de iterações é conhecido antecipadamente;
- oferece uma sintaxe clara e concisa para definir o início, a condição de parada e o incremento ou decremento da variável de controle.

### Flexibilidade:

- permite iterações com um número fixo ou variável de passos;
- a condição de parada pode ser definida por uma expressão lógica mais complexa.

# Estruturas de repetição – PARA/FAÇA

## Enunciado 1

Escreva um programa em pseudocódigo que

1. utilize a estrutura de repetição "Para" para realizar uma contagem progressiva de 1 a 10.
2. o programa deve exibir a mensagem "Contagem: [valor da variável de controle]" em cada iteração.



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – PARA/FAÇA

Solução em pseudocódigo:

1. INÍCIO
2. PARA contador DE 1 ATÉ 10 PASSO 1 FAÇA
3.     Escrever "Contagem: ", contador
4. FIM PARA
5. FIM



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – **PARA/FAÇA**

## Enunciado 2

1. Desenvolva um programa em pseudocódigo que utilize a estrutura de repetição "Para" para calcular a soma dos números pares de 2 a 20.
2. O programa deve exibir o resultado da soma.



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – PARA/FAÇA

Solução em pseudocódigo:

1. INÍCIO
2. soma  $\leftarrow$  0
3. PARA i DE 2 ATÉ 20 PASSO 2 FAÇA
4.     soma  $\leftarrow$  soma + i
5. FIM PARA
6. Escrever "A soma dos números pares de 2  
a 20 é: ", soma
7. FIM



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – **PARA/FAÇA**

## Enunciado 3

1. Crie um programa em pseudocódigo que utilize a estrutura de repetição "Para" para gerar a tabuada do 5.
2. O programa deve exibir cada multiplicação da tabuada.



Momento  
de **reflexão**

© Pexels



# Estruturas de repetição – PARA/FAÇA

Solução em pseudocódigo:

1. INÍCIO
2.   tabuada <- 5
3.   PARA i DE 1 ATÉ 10 PASSO 1 FAÇA
4.     resultado <- tabuada \* i
5.     Escrever tabuada, " x ", i, " = ", resultado
6.   FIM PARA
7. FIM



Momento  
de **reflexão**

© Pexels



Vamos  
fazer uma  
**atividade**

## Contagem regressiva:

**Enunciado:** Você está desenvolvendo um programa para simular a ignição de um foguete. O foguete tem um contador regressivo de ignição que começa em 10 e diminui a cada segundo. No entanto, a ignição só ocorre quando o contador atinge zero e um sinal de liberação é recebido. Além disso, o programa deve exibir mensagens específicas, dependendo do valor do contador.

Esta atividade deve ser executada e entregue no AVA com o nome de todos os participantes.



**20 min**



**individual**

- 1) Desenvolva um algoritmo em pseudocódigo, para simular essa situação, que utilize a estrutura de repetição "PARA" e outro algoritmo que utilize a estrutura de repetição "Para". O programa deve:
  - iniciar o contador em 10;
  - utilizar a estrutura "PARA" para exibir mensagens a cada segundo, indicando o tempo restante até a ignição;
  - utilizar a estrutura "Para" para realizar a contagem regressiva a partir do valor atual do contador até zero;
  - exibir uma mensagem específica quando a ignição ocorrer.
  
- 2) Esta atividade deve ser executada e entregue no AVA com o nome de todos os participantes.





O que nós  
**aprendemos  
hoje?**

© Getty Images

## Hoje desenvolvemos:

- 1** Conhecimento da estrutura de repetição “Para” (For), uma construção fundamental em programação, que permite a execução de um bloco de código enquanto uma condição específica permanece verdadeira;
- 2** Entendimento de que é importante que essa condição seja sempre trabalhada dentro da estrutura de repetição, caso contrário ela estará sempre em execução, e isso é um problema grave. Portanto, atente aos laços que possam ser infinitos;
- 3** Prática com atividade avaliada em AVA e com exercícios sem avaliação; solução de algoritmos com Estruturas de Controle de Repetição “PARA”, que contribuem largamente com o controle fino de repetições em algoritmos.





# Saiba mais

Será que você realmente precisa saber algoritmos e estruturas de dados para trabalhar com programação?

ALURA. *Algoritmos e estrutura de dados*, 2020. Hipsters #186. Disponível em:  
<https://cursos.alura.com.br/extra/hipsterstech/algoritmos-e-estrutura-de-dados-hipsters-186-a375>.  
Acesso em: 15 mar. 2024.

## Referências da aula

ALURA. *Algoritmos e estrutura de dados*, 2020. Hipsters #186. Disponível em: <https://cursos.alura.com.br/extra/hipsterstech/algoritmos-e-estrutura-de-dados-hipsters-186-a375>. Acesso em: 15 mar. 2024.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. *Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python*. Porto Alegre: Bookman, 2022.

Identidade visual: imagens © Getty Images

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**





## S12 – Aula 2 – Registro

### Contagem regressiva:

**Enunciado:** Você está desenvolvendo um programa para simular a ignição de um foguete. O foguete tem um contador regressivo de ignição que começa em 10 e diminui a cada segundo. No entanto, a ignição só ocorre quando o contador atinge zero e um sinal de liberação é recebido. Além disso, o programa deve exibir mensagens específicas, dependendo do valor do contador.

- Desenvolva um algoritmo em pseudocódigo, para simular essa situação, que utilize a estrutura de repetição "PARA" e outro algoritmo que utilize a estrutura de repetição "Para". O programa deve:
  - iniciar o contador em 10;
  - utilizar a estrutura "PARA" para exibir mensagens a cada segundo, indicando o tempo restante até a ignição;
  - utilizar a estrutura "Para" para realizar a contagem regressiva a partir do valor atual do contador até zero;
  - exibir uma mensagem específica quando a ignição ocorrer.

- Esta atividade deve ser executada e entregue no AVA com o nome de todos os participantes.

Condições de conclusão

Fazer um envio

## Resumo das Avaliações

Turmas separadas: 293566972 | 2ª SERIE BT MANHA ANUAL | 99 | JOAO CRUZ PROF

Oculto para estudantes	Não
Participantes	43
Enviado	0
Precisa ser avaliado	0



### Disciplina

Lógica, Algoritmo e Operação de Planilhas Eletrônicas 2º Bimestre

### Curso

Técnico em Ciência de Dados

### Ano letivo

2025



Retornar ao Sumário



# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**



# Lógica de programação e algoritmos

## Estruturas de controle: repetição

### Aula 3

Código da aula: [DADOS]ANO1C3B2S12A3

# Exposição



## Objetivo da Aula

Praticar o uso de estruturas de controle de repetição, com exercício avaliado pelo AVA.



## Competências da Unidade (Técnicas e Socioemocionais)

- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências;
- Compreender e dominar técnicas de manipulação de dados; extrair, transformar e carregar conjuntos de dados de diferentes fontes, garantindo a qualidade e a integridade dos dados; criar e compreender visualizações gráficas.



## Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens.



## Duração da Aula

50 minutos

Vamos  
fazer uma  
**atividade**

## Adivinhação aleatória:

**Enunciado:** Você está criando um programa de adivinhação em que o computador escolhe, aleatoriamente, um número entre 1 e 10 e o jogador tem que tentar adivinhar qual é o número escolhido. O jogo fornece dicas sobre se a tentativa do jogador está muito alta, muito baixa ou correta.

- 1) Desenvolva um algoritmo em pseudocódigo para simular esse jogo. O programa deve:
  - gerar, aleatoriamente, um número inteiro entre 1 e 10, que representará a resposta do computador;
  - solicitar ao jogador que faça uma tentativa de adivinhação;
  - comparar a tentativa do jogador com o número gerado aleatoriamente;
  - dar dicas ao jogador se a tentativa está muito alta, muito baixa ou correta;
  - permitir que o jogador faça múltiplas tentativas até adivinhar corretamente;
  - exibir uma mensagem indicando o número de tentativas necessárias para acertar.



**40 min**



**Em grupo**

- 2) Esta atividade deve ser executada com o nome de todos os participantes.



Vamos  
fazer uma  
**atividade**

## Adivinhação aleatória:



### Dica

O número gerado, aleatoriamente, exigiria o desenvolvimento de uma função; por isso, utilize o seguinte:  
`GerarNumeroAleatorioEntre1e10()`

A tentativa por parte do usuário, assim como o número aleatório gerado, pode ser: `LerTentativa()`



**40 min**



**Em grupo**



O que nós  
**aprendemos  
hoje?**

© Getty Images

## Hoje desenvolvemos:

- 1** Conhecimento da estrutura de repetição “Para” (*For*), uma construção fundamental em programação, que permite a execução de um bloco de código enquanto uma condição específica permanece verdadeira;
- 2** Entendimento que é importante que essa condição seja sempre trabalhada dentro da estrutura de repetição, caso contrário ela estará sempre em execução e isso é um problema grave. Portanto, atente aos laços que possam ser infinitos;
- 3** Prática com Atividade avaliada em AVA e com exercícios sem avaliação; solução de algoritmos com Estruturas de Controle de Repetição “PARA”, que contribuem largamente com o controle fino de repetições em algoritmos.



# Saiba mais

Aprenda a utilizar em Python as estruturas de controle de repetição.

ALURA. *Python para Data Science: primeiros passos*. 04. Estrutura de repetição. Disponível em: <https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/123750>. Acesso em: 15 mar. 2024.



# Referências da aula

ALURA. *Python para Data Science: primeiros passos*. 04. Estrutura de repetição. Disponível em: <https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/123750>. Acesso em: 15 mar. 2024.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. *Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python*. Porto Alegre: Bookman, 2022.

Identidade visual: imagens © Getty Images

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**