

Criação e simulação de algoritmos com repetição simples e aninhadas

Oficina de Programação para alunos do ensino médio

Representações de objetos

Um **modelo** é construído ao se **identificar padrões e atributos essenciais** de objetos.

Quais as **características essenciais** das figuras?



Juvenal



Cereja



Senhor Fabiano



Luka Maluka

disponível em: <https://loja.grupoa.com.br/eb-computacao-na-educacao-basica9786581334048-p1006668>

Desafio

“Quem sou eu?”



Senhor Fabiano



Luka Maluka



Juvenal



Cereja

Separados em duplas, cada aluno **escolhe um personagem**.

Escrever uma lista das **atributos** de seu personagem = {cor_de_cabelo, cor_olhos, gênero, usa_óculos, Qtde_acessorios, possui_bigode}

Cada aluno **escolhe uma característica** para responder sobre o seu próprio personagem;

Ganha quem **descobrir** primeiro o personagem escondido;

Classificação em conjuntos



Senhor Fabiano



Luka Maluka



Juvenal



Cereja

Diferentes características podem gerar diferentes agrupamentos:

Quantidade de personagens ?

Lista de nomes ?

QTDE Conjuntos separados por cor de cabelo?

Conjuntos separados por gênero?

Juvenal possui acessório no cabelo?

Cor do cabelo da Luka Maluka?

Classificação em conjuntos



Senhor Fabiano



Luka Maluka



Juvenal



Cereja

Os objetos podem ser classificados em conjuntos, que podem ser atômicos (números, palavras, valores-verdade) ou estruturados (registros, listas e grafos)

Quantidade de personagens = 4

Lista de nomes = {"Senhor Fabiano", "Luka Maluka", "Juvenal", "Cereja"}

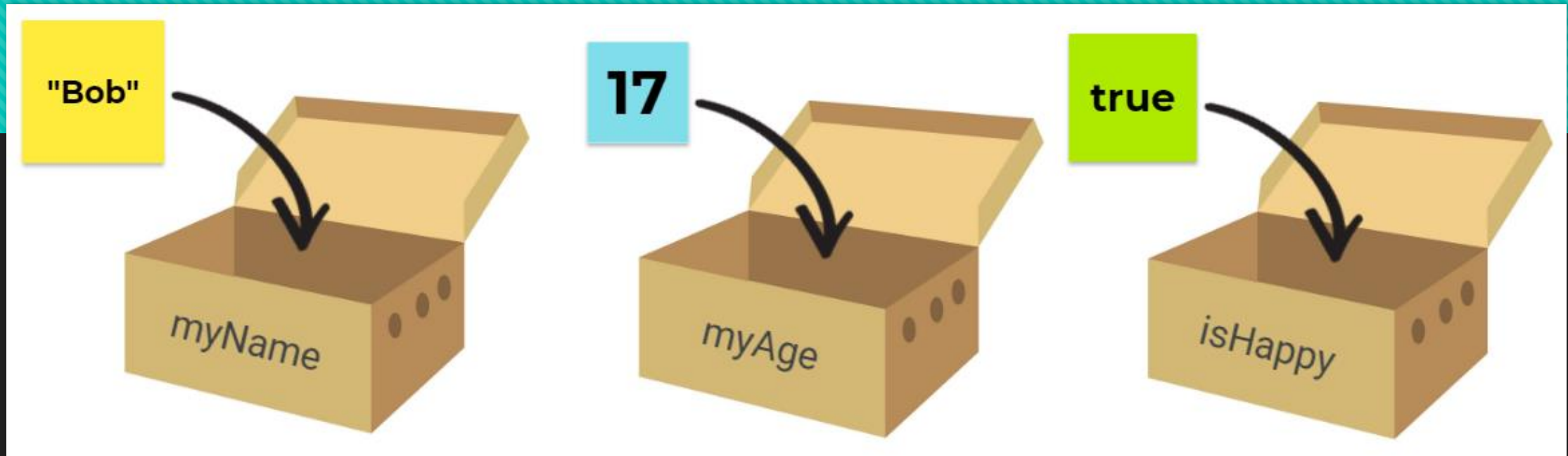
QTDE de Conjuntos separados por cor de cabelo = 3

Conjuntos separados por gênero = { {"Senhor Fabiano", "Juvenal"}, {"Luka Maluka", "Cereja"} }

Juvenal possui acessório no cabelo = Verdadeiro

Cor do cabelo da Luka Maluka = "loiro"

Tipos de dados em computação



- **inteiros:** números inteiros (0, 1, 2, 3...)
- **reais:** números inteiros e decimais (0; 1,2; 2,1; 3)
- **caracteres:** números reais, letras e outros símbolos ("pizza", "carro", "Vasco")
- **lógico:** comandos de VERDADEIRO ou FALSO

Tipos de dados



Senhor Fabiano



Luka Maluka



Juvenal



Cereja

Atributos de um personagem
= $\{ \text{cor_de_cabelo},$
 $\text{cor_olhos},$
 $\text{gênero},$
 $\text{usa_óculos},$
 $\text{Qtde_acessorios},$
 $\text{possui_bigode} \}$

inteiro

real

caracteres

lógico

IDENTIFICADORES

A utilização de nomes na programação é fundamental para se **identificar e referenciar** recursos num programa.

Usualmente estes são **palavras** cujo **significado** costuma se referir ao valor que este representa.

Isso é fundamental para que não haja confusão na interpretação.



Na cozinha, quando nos referimos a modelos de panelas diferentes (**frigideira, forma de bolo, caçarola**), indiretamente estamos **identificando**, através de **padrões e atributos essenciais**.

Seria como se **todos os utensílios tivessem nomes**.



Podemos imaginar uma variável como **utensílios e alimentos dispostos na bancada**.

Eles têm a capacidade de guardar certo **tipo de alimento**.

O nome da variável é como uma **etiqueta** colada na frente, identificando.

Variáveis

São **locais com um nome** dentro da **memória** do computador, criados em um algoritmo para **armazenar um determinado dado**;

Cada dado pode ser de um **tipo diferente** e, por isso precisamos mostrar ao nosso algoritmo logo no início o que esperar armazenar;

Entrada/Saída de dados:

Os algoritmos precisam interagir com o usuário.



Na cozinha de um restaurante, um **pedido**, juntamente com os **ingredientes necessários** para o seu preparo, são considerados os **dados de entrada**.

PREPARO DE
RECEITA



O **prato elaborado** resultante do pedido, juntamente com adereços como **bandejas, arranjos e outros artifícios** para embelezar o prato, representam a **saída**.

Entrada/Saída de dados

Em português, utilizamos o comando **leia()** para ler um dado enviado pelo usuário.

E o comando **escreva()** para retornar um dado para o usuário.

Se quisermos saber a altura de uma pessoa, devemos **escrever** para o usuário, **guardar a resposta** do usuário em uma variável chamada altura e **imprimir a resposta** para o usuário, como :

```
inteiro altura;
```

```
escreva ("Informe a sua  
altura");
```

```
leia (altura);
```

```
escreva ("Sua altura é:",  
altura);
```

Agora, podemos retomar o exemplo de algoritmo "Calcular idade":

Início

`inteiro idade;`

`inteiro ano;`

`escreva ("Em que ano você nasceu ?");`

`leia (ano);`

`idade = 2022 - ano;`

`escreva ("Sua idade é:", idade);`

Fim

Início do algoritmo

Declarando uma **variável** de nome `idade` com o tipo de dado inteiro

Recebe um dado do usuário

Lê o dado recebido pelo usuário e armazena o valor na variável `ano`

Variável `idade` **recebe o resultado** a operação de subtração entre o inteiro 2022 e o valor armazenado na variável `ano`

Retorna o valor da variável `idade`

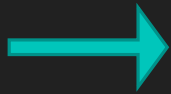
Fim do algoritmo

Roteiro para construir um algoritmo



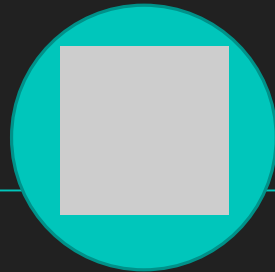
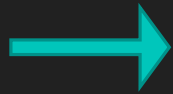
Entender o problema:

O que está sendo solicitado?



Identificar saídas:

O que deve ser calculado, processado, impresso?



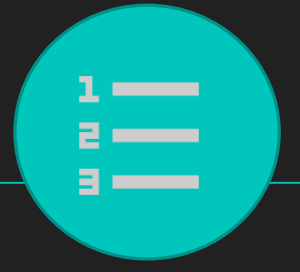
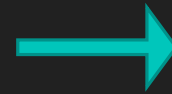
Identificar entradas:

Quais dados o usuário precisa fornecer para fazer os cálculos?



Identificar o Processamento

Quais cálculos necessários para que as entradas se transformem nos dados de saída?



Escrever os comandos em sequência:

Qual a ordem de execução?

Desafio:

Balança do Q-sabor

O restaurante a quilo K-Sabor cobra R\$39,90 por quilo de refeição.

Construa um algoritmo que leia o peso do prato montado pelo cliente (em quilos) e imprima o valor a pagar.

Assuma que a balança já desconta o peso do prato.



Não esqueça o **roteiro** para construir um algoritmo:

1. Entenda o problema;
2. Identifique as saídas;
3. Identifique as entradas;
4. Identifique o processamento necessário;
5. Escrever os comandos em sequências.



Assim como no preparo de um prato precisamos executar uma etapa repetidas vezes, podemos ensinar o **computador a fazer determinadas tarefas repetidas vezes**, sem preocupações com a fadiga.

As **estruturas de repetição** são muito utilizadas em desenvolvimento de softwares. Entender como funciona é muito importante para **resolver problemas que precisam executar tarefas repetidas vezes**.

Estruturas de repetição

São estruturas que permitem **executar mais de uma vez o mesmo comando** ou conjunto de comandos de acordo com uma **condição** ou com um **contador**.

Existem 3 estruturas de repetição básica para praticamente todas as linguagens de programação:

While (enquanto),

Do While (faça enquanto),

For (para)

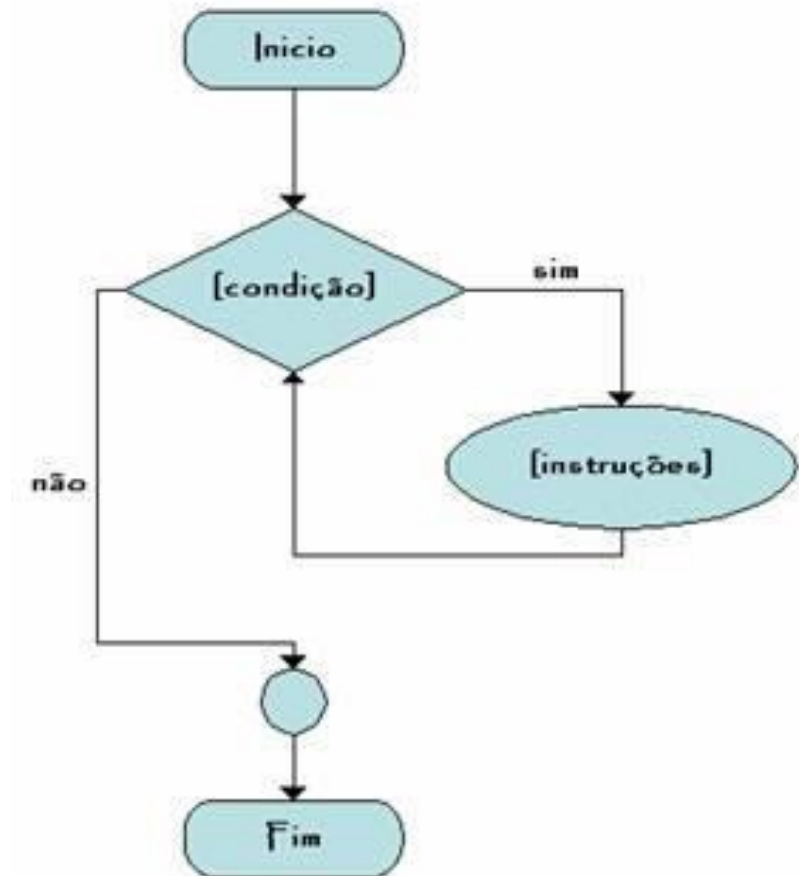


Enquanto (WHILE)

- Repete um bloco de código enquanto uma condição permanecer verdadeira
- Caso a condição seja falsa, os comandos dentro do while não serão executados e a execução continuará com os comandos após o while
- A repetição do while é controlada por uma condição que verifica alguma variável. Porém para que o while funcione corretamente é importante que essa variável sofra alteração dentro do while. Ex: um contador.
- Após entrar dentro da repetição, o bloco de comandos sempre será executado, mesmo que dentro do bloco a variável que está controlando a execução seja alterada.

Algoritmo "Par ou ímpar"

```
○ inicio
  inteiro contador
  contador = 0
  enquanto contador < 10 faz
    se (contador % 2 == 0)
      Escreva (contador , "É par")
    Senão
      Escreva (contador , "É ímpar")
      contador = contador + 1
  fimenquanto
fim
```



Faça enquanto

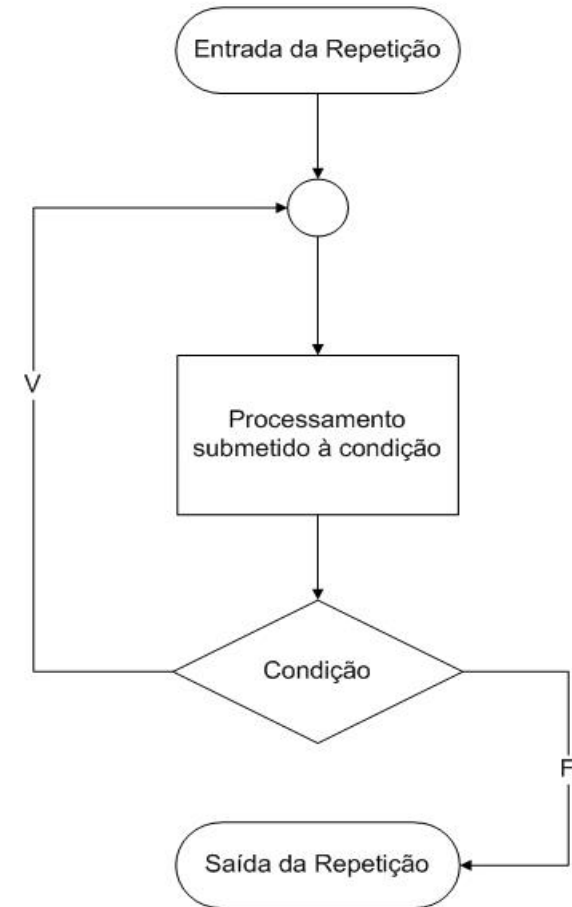
- Muito parecido com o while, porém tem uma diferença crucial: condição é verificada após executar o bloco de comandos.
- Há uma bloco de comandos e logo depois uma verificação. Assim caso a variável condicional for alterada dentro do bloco de comandos, isso afetará a validação da condição.
- A escolha entre while e do while é mínima, então dependerá do bom senso do programador, que optará pela estrutura que deixar o algoritmo mais simples e legível.

Algoritmo "Ler Senha"

Início

```
Inteiro senha_user, senha_cadastro  
senha_user = 0  
senha_cadastro = 100  
enquanto(1){  
  faça{  
    ler()senha_user)  
  }enquanto(senha_user != senha cadastro)  
  escreva("senha Válida)  
}  
fim
```

Implementação da estrutura repetitiva Faça - Enquanto



Para (FOR)

- O For é utilizado para executar um conjunto de comandos executado por um número X de vezes.
- É passada uma situação inicial, uma condição e uma ação a ser executada a cada repetição.
- Uma variável é inicializada com uma valor inicial.
- Essa variável é utilizada para controlar a quantidade de vezes em que o conjunto de comandos será executado.
- E ao final do conjunto de comandos a variável sempre sofrerá uma alteração, aumentando ou diminuindo de acordo com a lógica utilizada.

Algoritmo “Calcular Média”

Início

```
real nota, media, somaNotas = 0.0  
inteiro contador
```

```
para(contador = 1; contador <= 20; contador += 1){  
    escreva("Digite uma nota: ")  
    leia(nota)  
    somaNotas = somaNotas + nota  
}  
media = somaNotas / 20  
escreva("Média: ", media)
```

Fim

Exercícios:

Resolva os exercícios abaixo , utilizando a estrutura de repetição adequada

- Escreva um algoritmo para imprimir os 100 primeiros números pares.
- Escreva um algoritmo que receba a idade de 10 pessoas e calcule a média de idade.

Referências Bibliográficas

BAZILIO, C. Programando na cozinha. Disponível em: <https://carlosbazilio.gitbooks.io/programando-na-cozinha/content/pt-br/>. Último acesso em: 02/11/2022.

CAVALHEIRO, S.; FOSS, L.; AGUIAR, M.S; DU BOIS, A.R.; PERNAS, A.M.; REISER, R.H.S.; PIANA, C.F.B.; MAZZINI, A.R.A. Explorando o pensamento computacional para a qualificação do ensino fundamental. In: RAABE, A; ZORZO, A.; BLIKSTEIN, P. (org.). **Computação na educação básica**: fundamentos e experiências. Porto Alegre: Penso, 2020.336 p.

RAABE, A; ZORZO, A.; BLIKSTEIN, P. (org.). **Computação na educação básica**: fundamentos e experiências. Porto Alegre: Penso, 2020.336 p.

VILARIM, Gilvan de Oliveira. Algoritmos: Programação para Iniciantes — 3ª Edição. Rio de Janeiro: Editora Ciência, 2017