

# J A V A



**mentoriaT3ch**  
tecnologia, tendência & transformação.

## Sobre o autor

# Giovanni de Carvalho

Minha carreira iniciou em 2016 quando comecei a cursar um curso de manutenção de computadores pelo Instituto do Banco do Brasil, ali já tinha decidido que ia seguir na área da tecnologia. Entre idas e vindas em 2017 comecei a trabalhar em uma empresa de Telecom no setor administrativo e consequentemente iniciei uma graduação de Administração de Empresas na Universidade Nove de Julho. Após 2 anos de curso, estava atuando em uma multinacional e sendo destaque em sala de aula com palestras e projetos que foram patrocinados por grandes empresas como B3 e Coca-Cola.

Bom depois de tudo isso você deve estar pensando, que mais um jovem pode querer? Bom eu ainda quero muito mais e um destes enormes desejos era retornar a área de tecnologia.

Conquistei uma bolsa de estudos no centro de inovação do Facebook, para estudar programação Web, não pensei duas vezes! Pedi demissão do meu emprego para agarrar um dos meus sonhos. De lá para cá fui aluno do Senai, realizei diversos cursos e estou no último ano de uma graduação de Análise e Desenvolvimento de Sistemas. Trabalhei em um das Startups mais reconhecidas do país e hoje trabalho no maior Banco da América Latina.

Tenho experiência em engenharia de software com integração e modelagem de sistemas, transformação digital e modernização de plataformas legadas atuando com tecnologias como: Html5, Css3, Javascript, Saas, Bootstrap, Materialize, Angular, React, Spring Boot, Django, NodeJs, MySQL, Db2, Oracle, Mainframe, Mvc, Rest, Docker. Além de experiência em: administração de equipes, administração de processos gerando KPI's relatórios e análise de dados para melhorias contínuas.



“Feito é melhor do que perfeito” e é por isso que uma ideia pior do que a sua está fazendo sucesso

- Mark Zuckerberg





# Índice Geral

visão geral, introdução

ecossistema java

variáveis e constantes

operadores aritméticos

estrutura de decisão

estrutura de repetição

arrays

funções

programação orientada a objetos

desafio start java



# START

MÓDULO 1

visão geral, introdução

---

## JAVA

# Bórá entender um pouco sobre linguagens de programação



Basicamente todas informações são trabalhadas com dados /informações entre os seguintes pilares:

Entrada

Processamento

Saída

São envolvidos em uma sequência lógica de comandos com ordem definida com processamento finito.

## A linguagem java

O java é a linguagem mais utilizada no mercado, tendo responsabilidade no desenvolvimento de projetos famosos como: Uber, Ifood, Banco Itáu, Banco Bradesco, Pic Pay e etc;

## Linguagens de programação

- C++
- Java
- C#
- Ruby
- Python
- PHP
- JavaScript (NodeJS)

# START

MÓDULO 2

ecossistema java

---

# Ecossistema Java

JDK - kit de desenvolvimento Java para usar a linguagem Java com suas ferramentas.

Assim que baixarmos o JDK, podemos verificar se o Java está instalado com o código no terminal usando:

```
javac -version
```

```
java -version
```

## Download JDK

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

## Variáveis de ambiente

Após o término da instalação vamos configurar as variáveis do ambiente, abaixo estão os passos a serem seguidos.

- 1-Clique com o botão direito em cima do ícone "Meu Computador";
  - 2-Vá em "Propriedades";
  - 3-Selezione a aba "Avançado" (Windows XP) ou "Configurações avançadas do sistema", depois na aba "Avançado" (Windows 7);
  - 4-Clique no botão "Variáveis de ambiente";
  - 5-Clique no botão "Nova" em "Variáveis do sistema";
- Nome da variável: JAVA\_HOME  
Valor da variável: coloque aqui o endereço de instalação (o caminho tem que ser o mesmo especificado na Figura 10)  
C:\Arquivos de programas\Java\jdk1.5.0\_05  
Clique em OK



# JAVA

6-Clique novamente no botão Nova em Variáveis do sistema;

Nome da variável: CLASSPATH

Os valores da variável encontram-se abaixo, sempre insira um do lado outro sem espaços e com o ; (ponto e vírgula) no final.

```
1 ;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar;  
2 %JAVA_HOME%\lib\htmlconverter.jar;%JAVA_HOME%\jre\lib;%JAVA_HOME%\jre\lib\rt.jar;
```

Clique em OK

7-Selezione a variável PATH em Variáveis do sistema e clique no botão Editar.

Acompanhe os passos presentes na Figura 11.

Defina o valor dessa variável com o caminho da pasta Bin. No caso, pode-se utilizar a variável JAVA\_HOME previamente definida.

```
1 | ;%JAVA_HOME%\bin
```

Então confirme todas as telas no botão Ok

Para realizar os testes, após as configurações digite no terminal

```
1 | java -version
```



# S T - A R T

MÓDULO 3

**variáveis e constantes**



## JAVA

# Variáveis em Java

Espaço computacional para armazenar conteúdos de nossos programas.

### Parte técnica

- Pode começar com qualquer letra ou com \$ e \_
- Pode conter números, mas, não pode começar com números
- Não pode conter caracteres especiais como @, !, %, &
- Não pode conter acentos \*

Quando declaramos uma variável, temos que seguir um padrão de nomenclatura.

### Variáveis: nomes bons X nomes ruins

```
Double valorProduto = 100.0; // Melhor!  
Double valorproduto = 100.0; // Mais difícil ver a separação de palavras  
Double vlrPrd = 100.0; // Podemos confundir o que realmente significa  
Double vlrprd = 100.0; // Confusão e dificuldade de separar abreviações  
Double vp = 100.0; // Bem ruim  
Double x = 100.0; // Péssimo
```

## Diferença principal entre Tipo PRIMITIVO e Tipo NORMAL

```
public class TiposPrimitivos {  
  
    public static void main(String[] args) {  
  
        byte varByte = 1;  
  
        short varShort = 1;  
  
        int varInt = 1;  
  
        long varLong = 1;  
  
        float varFloat = 1f;  
  
        double varDouble = 1;  
  
        char varChar = 'A';  
  
        boolean varBoolean = true;  
    }  
}
```

String é o único tipo de dado que não tem tipo primitivo.

# S T \_ A R \_ T

MÓDULO 4

**operadores aritméticos**

---

## Operadores aritméticos

Use operadores aritméticos para realizar cálculos matemáticos em dados numéricos.

A seguinte tabela lista os operadores aritméticos na ordem de precedência na linguagem de transformação:

Operador	Tipo	Descrição
+ e -	ARITMÉTICO	Adição e subtração
* e /	ARITMÉTICO	Multiplicação e divisão
%	ARITMÉTICO	Módulo
> e <	RELACIONAL	Maior que e menor do que
>= e <=	RELACIONAL	Maior que e menor ou igual a
= e !=	RELACIONAL	Igual e diferente
=	ATRIBUIÇÃO	Simplemente, atribui algum valor para variável
++ e --	ATRIBUIÇÃO	Faz a operação - adição ou subtração - e, depois, a atribuição
*= e /=	ATRIBUIÇÃO	Faz a operação - multiplicação ou divisão - e, depois, a atribuição
%=	ATRIBUIÇÃO	Faz a operação - módulo - e, depois, a atribuição
&&	LÓGICO	Os dois lados do operador devem ser verdadeiros
	LÓGICO	Um dos lados deve ser verdadeiro
!	LÓGICO	Inverte o valor

## Operadores aritméticos

Adição, subtração, multiplicação, divisão e módulo.

### Precedência

1. Primeiro é multiplicação, divisão e módulo são avaliados primeiro. Havendo mais de um, eles serão avaliados da esquerda para direita;
2. Depois vem adição e subtração. Também, havendo mais de um, eles serão avaliados da esquerda para direita;

## Operadores relacionais

Igual, maior, menor

```
public class CalculadoraSimples {
    public static void main(String[] args) {
```

```
// > maior que
```

```
// < menor que
```

```
// == igual a
```

```
// >= maior igual
```

```
// <= menor igual
```

```
// != diferente de
```

```
// acima de 127 usamos o .equals
```

```
// sempre recebemos através de um Boolean
```

```
Integer n1 = 128;
```

```
Integer n2 = 128;
```

```
Boolean resposta = n1.equals(n2);
```

```
System.out.println("resposta: " + resposta);
```

```
}
```

```
}
```



# START

MÓDULO 5

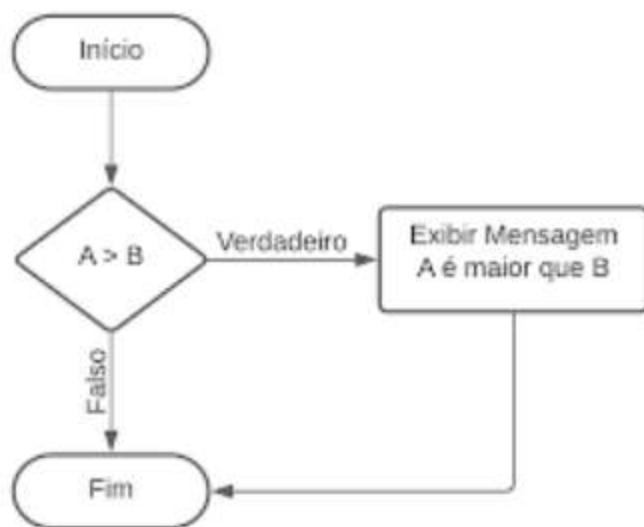
estrutura de decisão

---

# Estrutura de Decisão

A estrutura de decisão nos dá a possibilidade de decidir entre alguns caminhos que podemos seguir

## IF/ELSE



## Estrutura "if"

```
package decisao;

public class IfSimples {

    public static void main(String[] args) {

        Boolean movimentacao = true;
        Boolean tempoAbertura = true;
        Boolean nomeLimpo = true;

        Boolean liberaEmprestimo =
            movimentacao && tempoAbertura
                && nomeLimpo;

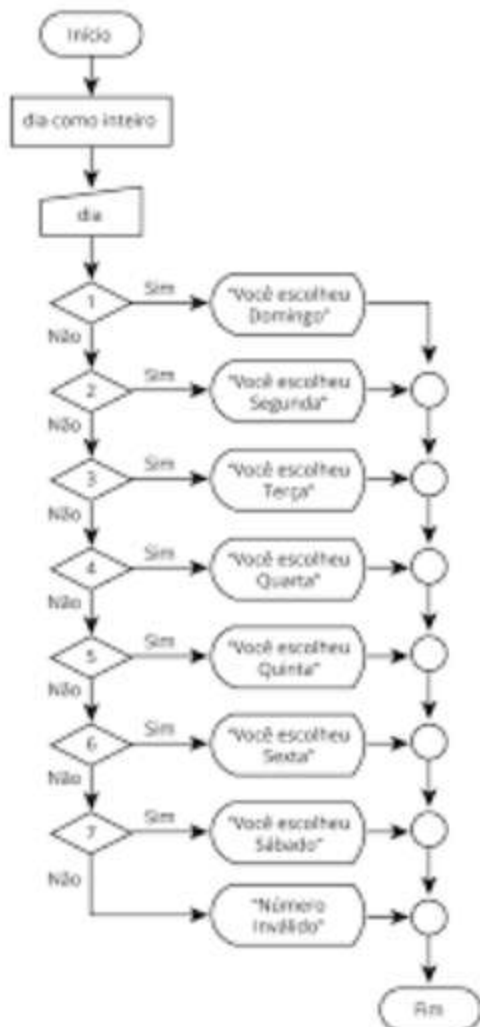
        if(liberaEmprestimo){
            System.out.println("sim, pode
            liberar");
        }
        else {
            System.out.println("não pode !");
        }

    }

}
```

# JAVA

## Utilizando o "if" encadeado



```
package decisao;
```

```
import java.util.Scanner;
```

```
public class IfEncadeado {
```

```
    public static void main(String[] args) {
```

```
        Scanner scan = new Scanner(System.in);
```

```
        System.out.print("digite o peso: ");
```

```
        Double peso = scan.nextDouble();
```

```
        Boolean semCategoria = peso <= 55;
```

```
        Boolean pesoLeve = peso <= 60;
```

```
        Boolean pesoMedio = peso >= 60 && peso
```

```
        <= 90;
```

```
        Boolean pesoPesado = peso >= 90 &&
```

```
        peso <= 120;
```

```
        if(pesoLeve) {
```

```
            System.out.println("lutador peso
```

```
            leve");
```

```
        }
```

```
        else if(pesoMedio) {
```

```
            System.out.println("lutador peso
```

```
            medio");
```

```
        }
```

```
        else if(pesoPesado) {
```

```
            System.out.println("lutador peso
```

```
            pesado");
```

```
        }
```

```
        else {
```

```
            System.out.println("lutador sem
```

```
            categoria");
```

```
        }
```

```
        scan.close();
```

```
    }
```

```
}
```

# S T - A R T

MÓDULO 6

estrutura de repetição

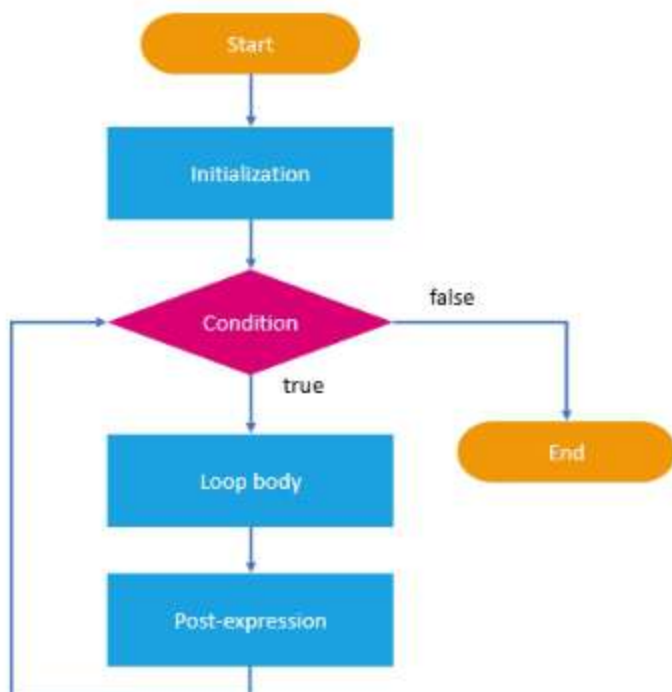
---



# Estrutura de Repetição

Atualmente existem 2 laços de repetição mais utilizados que são o:

For/While



## Introdução a iteração

Laços "for" e "while"

```
package repeticao;
```

```
public class LacoFor {
```

```
    public static void main(String[] args) {
```

```
        Double[] carrinhoCompras = new
        Double[] {40.0, 50.0, 60.0};
        Double total = 0.0;
```

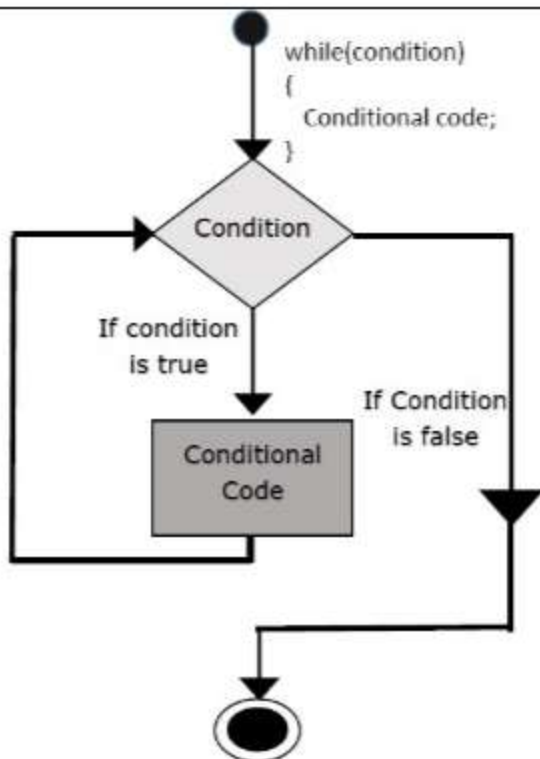
```
        for(int contador = 0; contador <
        carrinhoCompras.length; contador++) {
            System.out.println("Produto " +
            contador + "total " + total);
            total +=
            carrinhoCompras[contador];
        }

        System.out.println("total " + total);
    }
}
```

# JAVA

## Iterando com o laço "while"

Estrutura do While = enquanto  
De modo que enquanto a expressão  
booleana retornar falso ele executa.  
Quando for verdadeira ela sai do laço



```
package repeticao;
```

```
public class LacoWhile {
```

```
    public static void main(String[] args) {  
        // contador na parte de fora, pode  
        // ser reutilizado de qualquer parte do  
        // programa
```

```
        int contador = 0;
```

```
        // expressão booleana
```

```
        while(contador < 10) {  
            System.out.println(contador);
```

```
        // incremento
```

```
            contador++;
```

```
        }
```

```
    }
```

```
}
```

# START

MÓDULO 7

arrays

---

JAVA

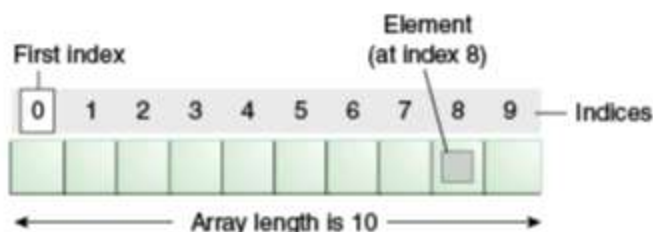
# Arrays

## Declarando vetores de 1 dimensão

Em programação de computadores, um arranjo (em inglês array) é uma estrutura de dados que armazena uma coleção de elementos de tal forma que cada um dos elementos possa ser identificado por, pelo menos, um índice ou uma chave.

Estrutura do Vetor Unidimensional

```
tipo[] (nomeVetor) = new tipo[] {  
informações, informações};
```



Cardápio de pizzaria

```
String[] cardapioPizza = new String[] {  
"calabresa", "atum", "mossarela"};
```

```
import java.util.Scanner;
```

```
public class VetorUnidimensional {
```

```
    public static void main(String[] args) {
```

```
        String[] cardapioPizza = new String[] {  
"calabresa", "atum", "mossarela"};
```

```
        System.out.println("Cardápio de  
pizzas :");
```

```
        for(int contador = 0; contador <  
cardapioPizza.length; contador++) {  
            System.out.println(contador + "  
sabor: " + cardapioPizza[contador]);  
        }
```

```
        Scanner scan = new Scanner  
(System.in);
```

```
        System.out.println("digite o número  
referente ao sabor da pizza:");
```

```
        Integer saborEscolhido =  
scan.nextInt();
```

```
        System.out.println("Sabor escolhido :  
" + cardapioPizza[saborEscolhido]);
```

```
        scan.close();
```

```
    }
```

```
}
```



# S T - A R T

MÓDULO 8

funções

---

# Funções

## Introdução a métodos

Uma função nada mais é do que uma sub-rotina usada em um programa. Na linguagem C, denominamos função a um conjunto de comandos que realiza uma tarefa específica em um módulo dependente de código. A função é referenciada pelo programa principal através do nome atribuído a ela.

Organização do código, facilidade de manutenção, ampliação do software e reaproveitamento código.

### Estrutura do método

```
modificador + tipo de retorno + nome do  
método( ) {  
    bloco de execução;  
}
```

chamar ele no código: nome metodo( );

```
static void posicaoInvalida() {  
    System.err.println("Posição  
inválida!");  
    System.exit(1);  
}
```

```
if (!posicaoValida) {  
    posicaoInvalida();  
}
```

# Funções

## Recebendo parâmetros

Uma função nada mais é do que uma sub-rotina usada em um programa. Na linguagem C, denominamos função a um conjunto de comandos que realiza uma tarefa específica em um módulo dependente de código. A função é referenciada pelo programa principal através do nome atribuído a ela.

Organização do código, facilidade de manutenção, ampliação do software e reaproveitamento código.

Quando passamos o parâmetro na Classe é chamado de: ARGUMENTOS.

Os parâmetros são usados para incluir ATRIBUTOS que não pertencem a Classe em que estamos usando.

### Estrutura do método com parâmetros

```
modificador + tipo de retorno + nome do  
método( Tipo do parâmetro que será  
recebido + nome da variável ) {  
    bloco de execução;  
}
```

chamar ele no código: nome método( +  
nome da variável do escopo MAIN );

```
static void lacoRepeticao(String[] vetor) {  
    for(int i = 0; i < vetor.length; i++) {  
        System.out.println "[" + i + "]" +  
vetor[i]);  
    }
```

```
String[] formasPagamento = new String[]  
{"Cartão", "Boleto"};
```

```
// recebe o nome do variável
```

```
lacoRepeticao(formasPagamento);
```

# START

MÓDULO 9

**programação orientada a  
objetos**

---



JAVA

# Programação orientada a objetos

## Introdução à programação orientada a objetos

Representar um objeto da vida real no meio computacional:

- atributos
- comportamentos

## Criando a primeira classe

Estrutura:

pacote principal

- Classe com o método main

- Classe que serão 'objetos'

conceitos abstraídos do projeto/sistema, ou seja abstrair o conceito em objeto

Quando criamos uma classe e passamos atributos, ela está pronta a se tornar um objeto só faltará instanciá-la.

Alem disso essa nova classe se torna um tipo, como se fosse uma string, um Double e por ai vai. a partir do tipo do objeto = (nome da classe), poderemos ter acesso aos atributos dentro da classe

JAVA

# Programação orientada a objetos

Como a maioria das atividades que fazemos no dia a dia, programar também possui modos diferentes de se fazer. Esses modos são chamados de paradigmas de programação e, entre eles, estão a programação orientada a objetos (POO) e a programação estruturada. Quando começamos a utilizar linguagens como Java, C#, Python e outras que possibilitam o paradigma orientado a objetos, é comum errarmos e aplicarmos a programação estruturada achando que estamos usando recursos da orientação a objetos.

Crie uma classe chamada produto e passe os seguintes atributos:

- nome
- preço
- quantidade

```
package poo;
```

```
public class Produto {
```

```
    String nome;  
    Double preço;  
    Integer quantidade;
```

```
}
```

Crie uma segunda classe que terá um método principal e receberá as funções que a classe produto irá realizar.

Também será instanciado o objeto nesta classe e setados os valores para execução.

```
package poo;
```

```
public class Principal {
```

```
    public static void main(String[] args) {
```

```
        Produto produto = new Produto();
```

```
        produto.nome = "celular j5";
```

```
        produto.preço = 450.00;
```

```
        produto.quantidade = 25;
```

```
        exibirQuantidadeEstoque(produto);
```

```
    }
```

```
    static void
```

```
    exibirQuantidadeEstoque(Produto  
    quantidadeEstoqueProduto) {
```

```
        System.out.println("O produto " +  
        quantidadeEstoqueProduto.nome + " tem "  
        +
```

```
        quantidadeEstoqueProduto.quantidade + "  
        unidade(s) em estoque");  
    }
```

# START

MÓDULO 10

**desafio start java**

---

JAVA

# Desafio

## #startJava

Nesta parte do desafio, iremos abrir a IDE eclipse e começar a desenvolver nosso programa.

### *#Desafio - Cálculo de gastos familiar*

Faça um programa que receba:

O valor da conta de luz

Conta de água

Conta de telefone

Escola do filho

Fatura do cartão

Gastos com supermercado

Mostre o gasto total que a família teve no mês.

### Código fonte

```
package operadores;

import java.util.Scanner;

public class DesafioCalculoGastoFamiliar {

    public static void main(String[] args) {

        Scanner scan = new
Scanner(System.in);

        System.out.print("digite o preço da
conta de LUZ: ");
        Double contaLuz =
scan.nextDouble();

        System.out.print("digite o preço da
conta de AGUA: ");
        Double contaAgua =
scan.nextDouble();

        System.out.print("digite o preço da
conta de TELEFONE/INTERNET: ");
        Double contaTelefone_Internet =
scan.nextDouble();
```



# Desafio

## #startJava

### Código fonte

```
System.out.print("digite o preço  
da conta de ESCOLA/FACULDADE: ");
```

```
Double contaEscola_Faculdade  
= scan.nextDouble();
```

```
System.out.print("digite o preço  
da conta de CARTÃO DE CRÉDITO: ");
```

```
Double contaCartao_Credito =  
scan.nextDouble();
```

```
System.out.print("digite o preço  
da conta de SUPERMERCADO: ");
```

```
Double contaSupermercado =  
scan.nextDouble();
```

```
Double totalMes = contaAgua +  
contaCartao_Credito +  
contaEscola_Faculdade + contaLuz +  
contaSupermercado +  
contaTelefone_Internet;
```

```
System.out.println("total gasto no  
mês é : " + totalMes);  
scan.close();  
}  
}
```

JAVA

# Desafio

## #startJava

### #Desafio - Verificar se a pessoa passou no concurso público

Crie um programa que vai receber as notas que uma pessoa tirou nas duas matérias de sua prova. Serão dois parâmetros para receber um para receber a nota de português e outro para receber as de matemática.

A prova, no total, vale 200 pontos - 100 para cada matéria. A nota mínima total para passar é igual ou maior que 150. Entretanto, o candidato não pode tirar menos do que 60 pontos em qualquer uma das duas matérias, ou seja, se tirar 59 em português e 100 em matemática (totalizando uma nota maior do que o total necessário que é 150) ele não conseguirá a vaga.

No final mostre para o candidato se ele conseguiu ou não.

### Código fonte

```
package decisao;

import java.util.Scanner;

public class DesafioConcursoPublico {

    static final Integer NOTA_CORTE = 60;
    static final Integer NOTA_APROVADO = 150;

    public static void main(String[] args) {

        Scanner leitora = new Scanner(System.in);

        System.out.print("digite a nota de matemática: ");
        Integer notaMatematica = leitora.nextInt();

        System.out.print("digite a nota de portugues: ");
        Integer notaPortugues = leitora.nextInt();
```

JAVA

# Desafio

## #startJava

### Código fonte

```
Boolean notaParcialMatematica =  
notaMatematica >= NOTA_CORTE;  
    Boolean notaParcialPortugues =  
notaPortugues >= NOTA_CORTE;  
    Boolean calculoMedia =  
(notaMatematica + notaPortugues) >=  
NOTA_APROVADO;
```

```
        if(notaParcialMatematica &&  
notaParcialPortugues && calculoMedia){  
            Integer mediaProva =  
notaMatematica + notaPortugues;  
            System.out.println("O aluno está  
aprovado, media final : " + mediaProva);  
        }  
        else {  
            System.out.println("o aluno está  
reprovado");  
        }  
  
        leitora.close();  
    }  
}
```

# Desafio

## #startJava

### #Desafio - Cadastrando as tarefas do dia

Crie um sistema onde o usuário vai cadastrar, utilizando o console, as 5 tarefas mais importantes do dia dele. Armazene cada uma dessas tarefas em um vetor e, no final, imprima as tarefas novamente.

```
package vetor;
```

```
import java.util.Scanner;
```

```
public class DesafioCadastroTarefas {  
  
    public static void main(String[] args) {
```

```
        String[] cadastroTarefas = new  
String[5];  
        Scanner scan = new  
Scanner(System.in);
```

```
        System.out.println("cadastre uma  
tarefa ");
```

```
        for( int contador = 0; contador <  
cadastroTarefas.length; contador++ ){  
            System.out.print(contador + "°  
tarefa: ");  
            cadastroTarefas[contador] =  
scan.nextLine();  
        }
```

```
        System.out.println(" As tarefas são: ");
```

```
        for( int contador = 0; contador <  
cadastroTarefas.length; contador++ ){  
  
System.out.println(cadastroTarefas[contado  
r]);  
        }  
  
        scan.close();  
    }  
}
```





# Obrigado

Queremos agradecer todo seu empenho e disposição em participar deste projeto que é a mentoria tech uma comunidade construída de alunos para alunos com grandes ambições e objetivos dentre eles temos:

- Fazer com que os alunos alcancem altos níveis em suas formações.
- Mentoria de alunos experientes para apoiar e moldar os alunos mais novos.
- Agregar valor na busca de seu primeiro emprego.
- Produzir e compartilhar conteúdo de qualidade feito por alunos com foco em alunos.
- Disseminar e promover acessibilidade e igualdade em nossos conteúdos ajudando assim toda cota de alunos.
- Realizar encontros com outras comunidades e eventos que gerem impactos positivos como networking, experiências, preposições técnicas entre outros.

# Busque nossas redes sociais



+55 (11) 9 61985346



contatomentoriatech@gmail.com



@mentoriatech



[www.mentoriatech.com.br](http://www.mentoriatech.com.br)

## Bootcamp



ACESSE AGORA

Siga nossas mídias sociais  
[@mentoriatech](#)