



# Projeto Integrador I

*Prof. Dr. Eliseu LS*

# APRESENTAÇÃO DOCENTE

***Prof. D.r Eliseu Lemes da Silva***

## ***FORMAÇÃO ACADÊMICA ATUAL:***

1. *Pós-doutorado em Educação ênfase em Psicologia Cognitiva (UFLO/AR)*
2. *Título de Doutor em Educação ênfase em Humanidades e Artes "Tese Ensino Superior Tecnológico" (UNR/Mercosul/Argentina)*
3. *Título de Doutor em Educação ênfase em Inclusão Social - UFMG (Brasil)*
4. ***Especialização em Engenharia da Computação (UFU/MG)***
5. ***Licenciatura em Matemática (UMESP)***
6. ***Graduação em Processamento de Dados (UNESP)***
7. *Bacharelado em Direito (IME/SP)*

## ***ATIVIDADES PROFISSIONAIS ATUAIS:***

1. ***Coordenador de Curso Superior de Tecnologia em ADS (CEETEPS/FATEC-SP)***
2. ***Professor Titular na área de Tecnologia da Informação (CEETEPS/FATEC-SP)***
3. ***Professor Titular na área de Tecnologia da Informação (SENAC/SP)***
4. ***Pesquisador / Escritor de literatura sobre Educação.***

**Curriculo Lattes: <http://lattes.cnpq.br/9300104925024179>**

# EMENTA DO CURSO

Propor o desenvolvimento de um projeto interdisciplinar de desenvolvimento de sistemas. Apresentar os aspectos fundamentais de programação visual utilizando a linguagem de programação java para a construção de um **sistema totalmente funcional**, incluindo persistência em banco de dados. Introduzir conceitos de orientação a objetos. Apresentar os elementos necessários para a prototipação e codificação das funcionalidades.

## ***BIBLIOGRAFIA BÁSICA :***

DEITEL, P. J. **Java**: como programar. São Paulo: Pearson Education, 2016.

## ***BIBLIOGRAFIA COMPLEMENTAR :***

EVANS, B. J.; FLANAGAN, D. **Java in a Nutshell**: O'Reilly, 2014.

FREEMAN, E.; ROBSON, E. **Head First Design Patterns**: O'Reilly, 2014.

## **Links e outras fontes :**

- ***Notas de aula Dr. Eliseu LS: Slides, exercícios de aprendizagem e vídeo-aulas.***
- ***<https://www.devmedia.com.br/guia/linguagem-java/38169>***
- ***[www.oracle.com](http://www.oracle.com)***
- ***<https://dev.mysql.com/downloads/connector/j/>***

# METODOLOGIA

## Entrega dos Tarefas / ADOs;

1. Recursos obrigatórios: Caderno, Computador e Internet;
2. As entregas ou códigos dos programas serão entregues pela ferramenta adotada na disciplina ( **Blackboard** ) sempre dentro de um arquivo no formato MS-WORD contendo - Nome dos autores, enunciado, código e prints de execução ou de diagramas usados;
3. Cada atividade irá receber uma pontuação de 2 até 4 pontos com prazo de entrega de até uma semana, salvo exceções, tarefas atrasadas só valem até 2 pontos e não serão aceitas após 3 dias de atraso;
4. **Ao final do semestre** será atribuída uma nota entre 0 (Zero) e 10 (Dez) conforme percentual de pontos obtidos nas ADOs que será usada para composição da média final junto com as demais avaliações;

## Dinâmica da Aula:

1. ***Aula interativa com modelos, tutoriais e exercícios de aprendizagem:*** Nesta parte, o professor apresenta o princípio de funcionamento de cada comando e também os conceitos de lógica, em seguida exibe no data show, de forma dinâmica ou estática, um exercício de aprendizagem contendo os conceitos de lógica e comandos que foram ensinados.
2. ***Prática de ADOs / avaliação contínua :*** Nesta parte da aula, caberá aos alunos a prática dos comandos e conceitos explicados pelo professor, caberá a ele (a) o esforço para fazer as atividades propostas podendo recorrer ao professor em qualquer momento durante a aula presencial para sanar suas dúvidas nos códigos, etc.
3. ***Forma de correção de ADOs / Exercícios de Fixação:*** O processo de correção das atividades se dará de forma individualizada e pessoal durante o processo e também após a entrega que deverá ser feita através da ferramenta adotada pela disciplina.

# CORREÇÃO DAS ADOs

## **a) A nota do código das avaliações semanais perderá pontos quando :**

1. o enunciado não for atendido na íntegra;
2. o código não código não contiver pacotes, bibliotecas ou a diretivas necessárias;
3. o código não não funcionar;
4. Número de sub rotinas ou métodos estiver diferente do que foi pedido no enunciado;

## **a) O código dos métodos sub rotinas será considerado totalmente errado quando:**

1. A lista de parâmetros ou argumentos não estiver de acordo com o enunciado;
2. Os parâmetros e argumentos estiverem incorretos ou declarados sem necessidade;
3. contiver fórmulas erradas ou montadas de forma incorreta

# Cálculo da Média Final

**Cada atividade terá uma nota entre 0 (zero) e 10 (Dez).**

**( 10% da nota final ) ADOs:** Tutoriais, projetos práticos, avaliação Contínua ao final totalizam até 10% da média final da disciplina;

**( 10% da nota final ) Documentação/Especificação :** Pré-projeto escrito do software com padronização ABNT, contendo a descrição do software, objetivos, especificações, requisitos, diagramas, prototipagem, modelo de dados, etc. Esta atividade totaliza até 10% da nota final da disciplina;

**( 80% da nota final ) Software :** Software com CRUD funcional em Java com acesso a banco de dados. (Persistência de Dados), totaliza até 80% da nota final da disciplina;

$$\text{MÉDIA} = (\text{Especificação} * 0.1 + \text{Software} * 0.8 + \text{ADOs} * 0.1 ) / 10;$$

**ATENÇÃO :** Para a aprovação o aluno deverá obter no mínimo a média 6 sendo que a disciplina não contempla avaliações substitutivas ou de recuperação.

# Entrega do PI e Atividades

**Atividades ADOs** - Entregas semanais ou quinzenais sempre com datas fixadas na plataforma adotada ( **Blackboard** ), as ADOs sempre serão apresentadas no último slide do PDF da aula.

**Especificação e Software** - A apresentação final do PI-II, bem como a entrega da especificação do projeto escrito será entre a 15 e 16 semana que ocorrerá no laboratório designado para a disciplina.

**APRESENTAÇÃO:** Os membros do grupos devem estar presentes adequadamente trajados evitando exageros e caso algum membro se ausente sem justificativa a nota final do trabalho poderá ter reduções. Após a apresentação o código do software zipado bem como o projeto escrito deverá ser entregue estritamente pelo Blackboard com limite na 15 semana do semestre.

**ATENÇÃO:** As consultorias e vistorias do professor com os grupos de PI-II ocorrerão em todas as aulas até a data de **16-5-2024**.



**(Java)**  
**Projeto Integrador II**  
**Aula 1**

*Prof. Dr. Eliseu LS*





**(Java)**  
**Algoritmos I**  
**Aula 1**

*Prof. Dr. Eliseu LS*



***(Java)***  
***Introdução à***  
***Lógica Computacional***

*Prof. Dr. Eliseu LS*

# CONCEITO DE ALGORITMO

Um algoritmo pode ser definido como um conjunto de regras (instruções), bem definidas, para solução de um determinado problema. Segundo o dicionário Michaelis, o conceito de algoritmo é a utilização de regras para definir ou executar uma tarefa específica ou para resolver um problema específico (Carvalho, 2007).

Trata-se de uma sequência de ações ou passos lógicos, para se resolver um determinado problema ou desafio. Cada ação é chamada de comando e dependerá de decisões que vão sendo tomadas com a finalidade de se alcançar a resolução do problema em questão.

# REPRESENTAÇÃO DE ALGORITMOS

- a) **Linguagem descritiva:** Seria o uso da linguagem humana para descrever os passos de um algoritmo, por exemplo, o enunciado detalhado de uma tarefa, uma receita de bolo, como trocar o pneu de uma carro, como trocar uma resistência de chuveiro, etc.
- b) **PseudoCódigo:** É quando se usa comandos em linguagem natural ou nativa do programador, em nosso caso, utiliza-se o Português estruturado, famoso PORTUGOL.
- c) **Diagrama de Blocos:** Diagrama de Blocos é um conjunto símbolos que são utilizados para descrever um algoritmo computacional, cada símbolo representa um comando específico dentro algoritmo, que deve possuir somente um INÍCIO é um FIM.
- d) **Programa Computacional:** Trata-se de um conjunto de comandos e ações escritos em uma linguagem conhecida do ser humano e do computador. Um programa poderá ser desenhado em linguagens, tais como, cobol, c++, android, java, php, python, etc.

# ALGORITMOS COMPUTACIONAIS

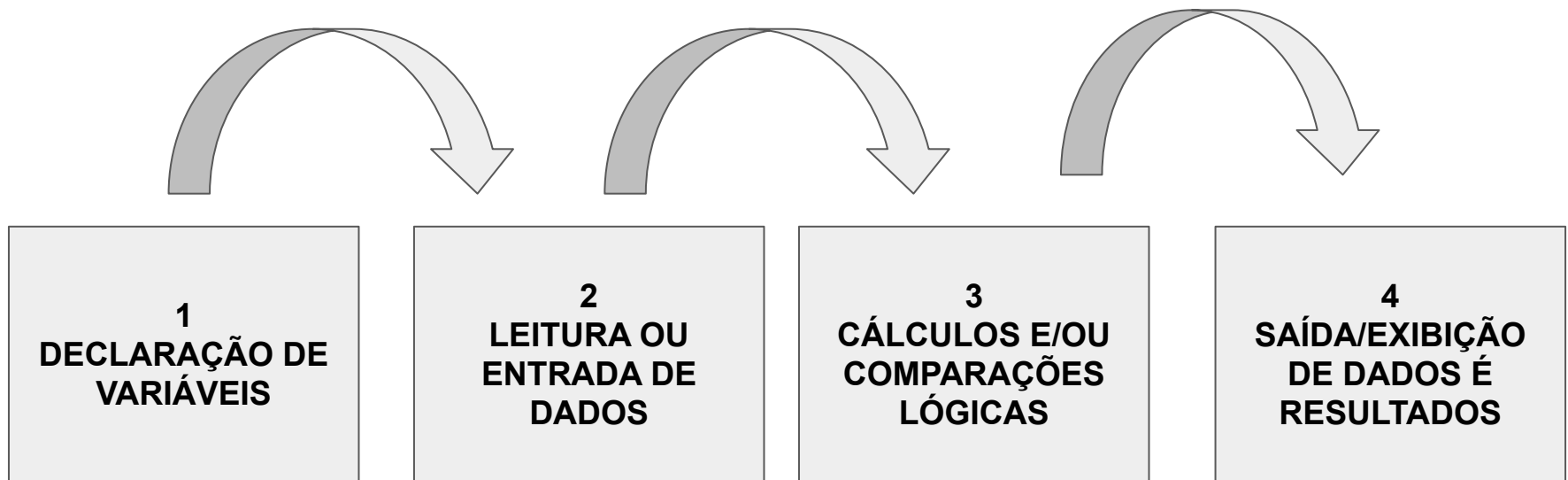
Envolvem o uso de computadores para solucionar problemas, requerem o uso da **lógica computacional ou lógica de programação** que consiste no conjunto de comandos e instruções que serão traduzidos para a língua inglesa de acordo com a **linguagem de programação** utilizada.

Uma linguagem de programação seria a linguagem que o computador poderá ler e traduzir para sua linguagem nativa, que é sempre **binária**.

Ao ser traduzido para uma linguagem de programação, um ou vários algoritmos irão formar o **Programa**, o texto que contém os comandos e instruções de um programa recebe o nome de **Código Fonte ou Arquivo Fonte**.

# PARTES DE UM ALGORITMO/CÓDIGO FONTE

O código fonte de um algoritmo computacional é dividido em 4 (quatro) partes:



*As 4 (quatro) fases de um algoritmo computacional*

# DESCRIÇÃO DAS PARTES

**1 DECLARAÇÃO DE VARIÁVEIS:** Primeira parte de um algoritmo, onde deverão ser criadas as variáveis e constantes que serão utilizadas nos cálculos. Ex.: x , y , Nome1, x\_2, endereCo, saida, resultado, Media; Constantes: pi = 3.14; gravidade = 9.8

**2 LEITURA/ENTRADA DE DADOS:** Nesta parte, os valores de cada variável de ENTRADA, será digitado pelo usuário, com isso, os resultados das variáveis de saída serão calculados de acordo com os valores iniciais de cada variável de entrada.

**3 PROCESSAMENTO:** Nesta parte do algoritmo/programa, serão feitos os cálculos, fórmulas e as comparações lógicas, enfim todo o processamento para gerar os resultados esperados pelo programador e cada variável de saída receberá o seu valor como resultado dos processos e fórmulas envolvidas.

**4 SAÍDA DE DADOS:** Trata-se da fase de exibição dos resultados finais, ou seja, apresentação ou impressão das variáveis de saída, que são as variáveis que irão conter os resultados dos cálculos efetuados pelo código do programa ou algoritmo.

# **Simbologia e Diagramas**



# PORTUGOL - PORTUGUÊS ESTRUTURADO

Para o estudo de lógica, o uso do Português Estruturado é indicado quando não há interesse no aprendizado de uma linguagem de programação.

```
var
  x: inteiro;
  a: inteiro;
  b: inteiro;
início
  escreva("Informe o primeiro valor: ");
  leia(a);
  escreva("Informe o segundo valor: ");
  leia(b);
  x = a + b;
  escreva("o resultado da soma é: ", x);
fim
```

} Declaração de variáveis

} Bloco de Instruções

**Modelo de Portugol**

# DIAGRAMAS UTILIZADOS

Para representar a lógica de programas, os diagramas mais comuns utilizados são: o Diagrama de Chaplin e o Diagrama de Blocos.

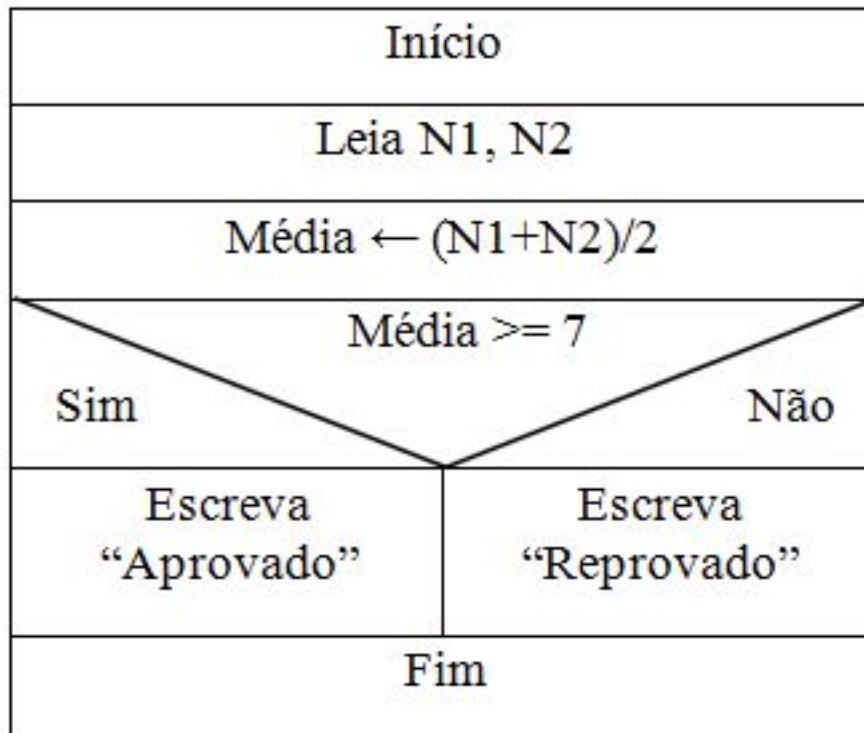


Diagrama de Chaplin

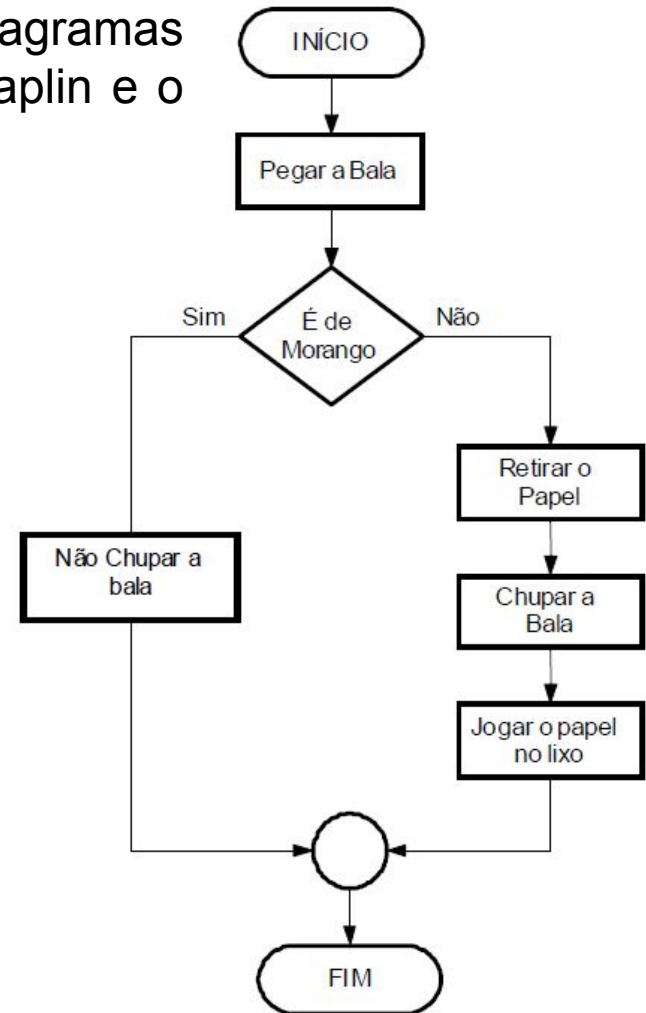


Diagrama de Blocos


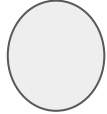
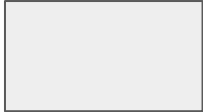




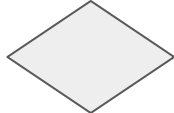
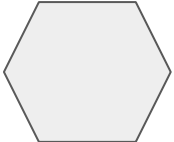

# MODELO P4 (Quadro Resumo)

Considere o algoritmo para calcular o **MÉDIA** aritmética entre duas notas, **NOTA1** e **NOTA2**, onde as notas serão informadas pelo usuário. .

**FÓRMULA:  $MEDIA = (NOTA1 + NOTA2) / 2$**

INSTRUÇÕES	QUADRO RESUMO DAS 4 PARTES
1. Nomear e atribuir valores iniciais para <b>todas as variáveis e constantes</b> do problema	1. $MEDIA = 0$ , $NOTA1 = 0$ , $NOTA2 = 0$
2. Listar variáveis de <b>LEITURA</b> , ou seja, aquelas que serão informadas pelo usuário	2. $NOTA1$ , $NOTA2$
3. Listar os cálculos/fórmulas do problema	3. $MEDIA = (NOTA1 + NOTA2) / 2$
4. Listar variáveis de <b>SAÍDA</b> , ou seja, os RESULTADOS das fórmulas ou cálculos	4. $MEDIA$

# SIMBOLOGIA PROPOSTA (Diagrama de Blocos)

1 <b>ELIPSE</b> : Usado para o <b>INÍCIO</b> e <b>FINAL</b> do diagrama.	
2 <b>CONECTOR</b> : Quando duas ou três linhas forem para um mesmo local, deve-se utilizar o Conector.	
3 <b>VARIÁVEIS</b> e <b>CONSTANTES</b> : As variáveis e constantes devem ser criadas dentro de um <b>RETÂNGULO</b> .	
4 <b>LEITURA DE VARIÁVEIS</b> : Para Leitura de variáveis pode-se usar um <b>PARALELOGRAMO</b> ou um destes três símbolos.	  
5 <b>CÁLCULO E DECISÃO</b> : Para fórmulas e comandos utilizar o <b>RETÂNGULO</b> , para decisão ( IF else ) e Laços ( WHILE ) utilizar o <b>LOSANGO</b> , <b>EXÁGONO (FOR)</b>	  
6 <b>SAIDA/RESULTADOS</b> : Para mostrar as saídas ou resultados do programa utiliza-se o <b>PARALELOGRAMO</b> ou um destes três símbolos.	

# APRENDIZAGEM - Enunciados

**Programa 1:** Utilizando os símbolos propostos, criar um diagrama de blocos que mostre o valor da MÉDIA entre duas notas digitadas pelo usuário. (NOTA1 e NOTA2).

$$\text{Fórmula: } \text{MEDIA} = (\text{NOTA1} + \text{NOTA2}) / 2$$

**Programa 2:** Utilizando os símbolos propostos, criar um diagrama para exibir RESULTADO, sendo que o valor de RESULTADO será 10 vezes o valor de PI = 3.14159.

$$\text{Fórmula: } \text{RESULTADO} = 10 * \text{PI}$$

# Documentos - Programa 1

## Diagrama 1

### Quadro Resumo (QR)

1. NOTA1, NOTA2, MEDIA
2. NOTA1, NOTA2
3.  $MEDIA = (NOTA1 + NOTA2) / 2$
4. MEDIA

### ALGORITMO DESCRITIVO (AD)

1 *Configurar o idioma*

2 *Apagar a tela*

3 Criar variáveis:

NOTA1=0, NOTA2=0, MEDIA =0

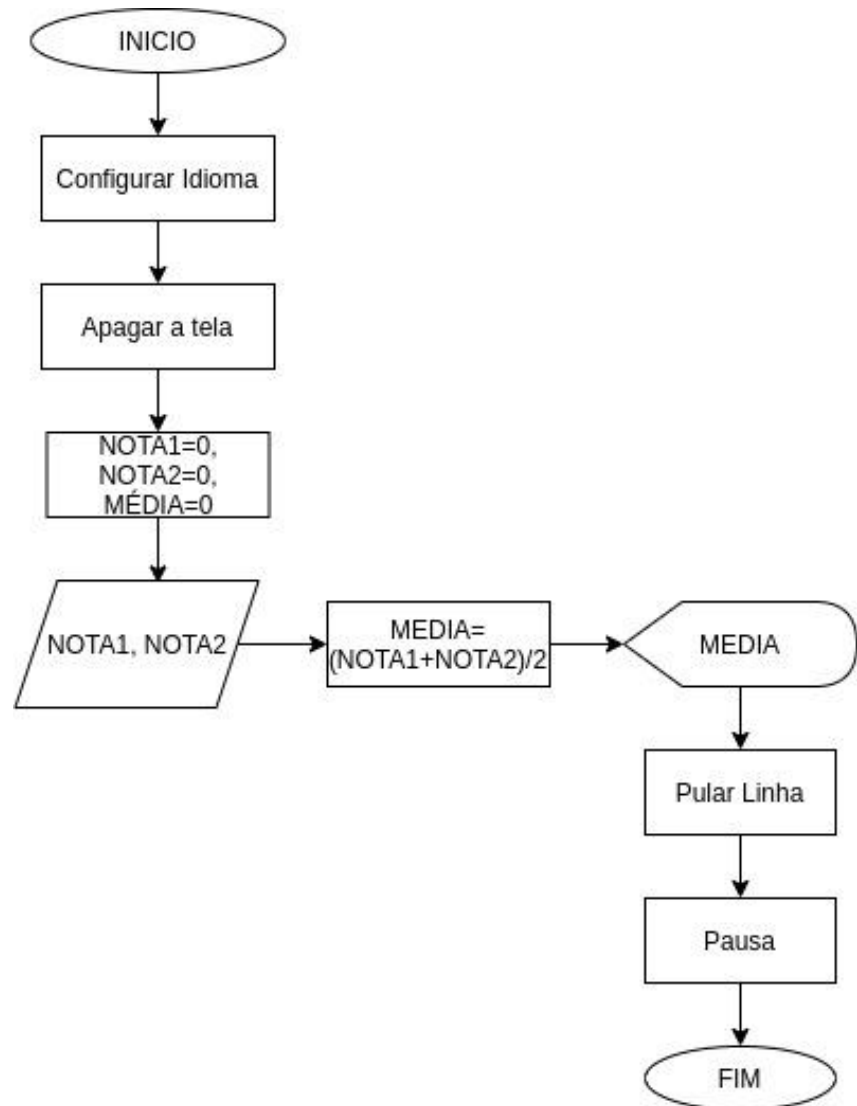
4 Ler as variáveis : NOTA1, NOTA2

5 Calcular  $MEDIA = (NOTA1 + NOTA2) / 2$

6 Exibir a MEDIA

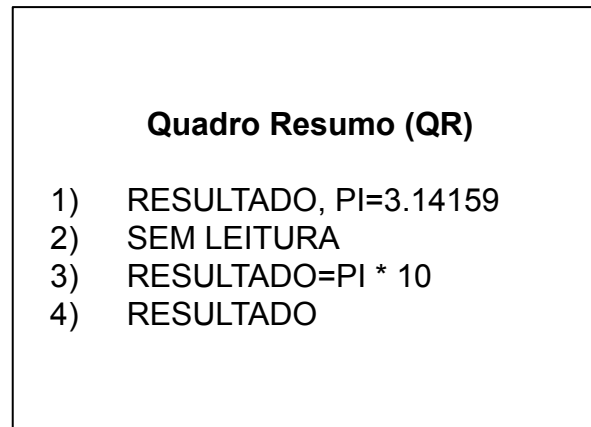
7 *pular uma linha*

8 *dar uma pausa*



# Documentos - Programa 2

## Diagrama 2



### ALGORITMO DESCRITIVO (AD)

1 *Configurar o idioma*

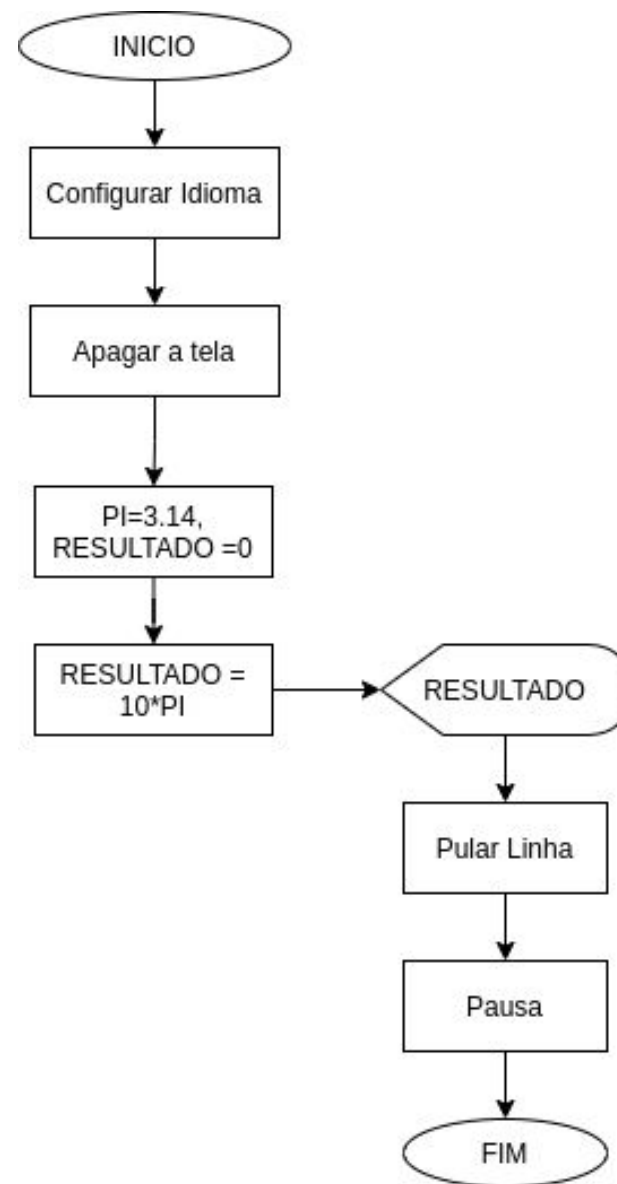
2 *Apagar a tela*

3 criar a constante  $PI = 3.14$  e a variável RESULTADO;

4 calcular o  $RESULTADO = PI * 10$ ;  
5 exibir RESULTADO;

6 *Pular linha*

7 *Dar uma pausa;*



# Variáveis, Constantes & Quadro Resumo

**VARIÁVEL:** Trata-se de um espaço de memória que sofrerá alterações durante a execução de um algoritmo, a variável é declarada por um NOME e seu tipo de dado (datatype).  
Exemplos: `double salario;` `int idade;` `char sexo= 'F';`

**CONSTANTE:** Trata-se de um espaço de memória que não sofrerá e nem permitirá alteração de seu valor até o final da execução do programa. Também deve ser informado em sua declaração seu nome e seu tipo de dado (datatype). `pi=3.14159;` `g= 9.8;`  
`v=340;`

**VARIÁVEL DE LEITURA ou de ENTRADA de dados:** São as variáveis que o usuário deverá informar para que o código possa fazer os processos ou cálculos do algoritmo.

**VARIÁVEL DE SAÍDA de dados:** São as variáveis que irão armazenar os resultados dos cálculos que serão exibidos sempre no final do código ao usuário.



## ADO 1 / Laboratório (Primeira Parte)

Fazer o quadro resumo de variáveis (QR) que contém 4 partes de um algoritmo de acordo com o slide número 14, depois faça o Diagrama de Blocos (DB) e finalmente faça o Algoritmo Descritivo (AD) de acordo com os slides 15 - 16 para os enunciados A, C, D, E e H do exercício (7) das páginas 25 e 26 do livro de Exercícios: Estudo Dirigido de Algoritmos Manzano.

**NOTA:** *Entregar em arquivo "ÚNICO" no formato DOCX ou PDF, enunciado, diagramas e/ou código dentro do documento enviado através do ambiente virtual de aprendizagem - BlackBoard - dentro da data estipulada no mesmo.*