

# Introdução à programação



# André Felipe Bürger



Tech Lead e Desenvolvedor Full Stack – Pública Tecnologia



## Formação acadêmica

Graduado em Sistemas de Informação - FURB

Especialização em Tecnologias Para o Desenvolvimento de Aplicações WEB - FURB

MBA em Inovação e Gestão Ágil de Projetos - FAE



## Matérias lecionadas

- Algoritmos e Estrutura de Dados;
- Introdução à Programação;
- Programação Orientada à Objetos;



/afbuerger



# Plano de ensino

## **Dividido em 7 unidades:**

1. Fundamentos da programação de computadores
2. Construção de algoritmos
3. Introdução à linguagem de programação
4. Comandos de controle de fluxo: Seleção
5. Comandos de controle de fluxo: Repetição
6. Tipos estruturados
7. Introdução à orientação à objetos



# 1. Fundamentos da programação de computadores

- Solução de problemas
- Técnicas para representação da solução

## 2. Construção de algoritmos

- Dados e Tipos
- Comandos e Instruções



# 3. Introdução à linguagem de programação

- Introdução de uma IDE;
- Características da linguagem de programação;
- Tipos de dados;
- Palavras reservadas;
- Operadores;
- Comandos de entrada e saída;
- Método “main” e conceitos de subprograma;
- Passagem de parâmetros por valor e referência;
- Retorno da função;



## 4. Comandos de controle de fluxo: Seleção

- Simples: se (if)
- Encadeada: se-senão (if – else)
- Multipla: escolha (switch – case)

### Prova 01 (Individual – Teórica e prática)

- Unidade 1;
- Unidade 2;
- Unidade 3;
- Unidade 4;



## 5. Comandos de controle de fluxo: Repetição

- Enquanto (while);
- Para (for);
- Faça enquanto (do – while);

## 6. Tipos estruturados

- Características dos tipos estruturados;
- Vetores;

## Prova 02 (Individual – Teórica e prática)

- Unidade 5;
- Unidade 6;



# 7. Introdução à orientação a objetos

- Classes e objetos;
- Atributos;
- Mensagens e métodos;
- Encapsulamento;
- Construtores;

## Trabalho final (Em dupla – Implementação + Apresentação) \*avaliado individualmente

- Unidade 1;
- Unidade 2;
- Unidade 3;
- Unidade 4;
- Unidade 5;
- Unidade 6;





# Procedimento de avaliação

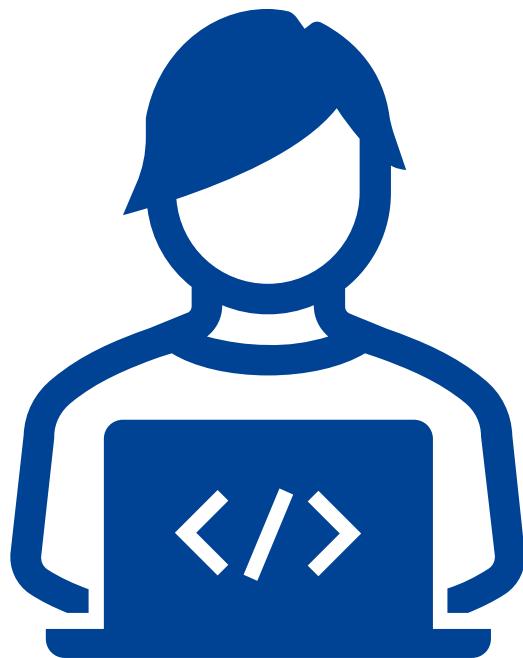
- Prova 1 \* (0,20) – Peso 2;
- Prova 2 \* (0,25) – Peso 2,5;
- Trabalho final \* (0,35) – Peso 3,5;
- Exercícios \* (0,20) – Peso 2;

## Exemplos:

- |                             |                     |                             |                     |
|-----------------------------|---------------------|-----------------------------|---------------------|
| • Prova 1:                  | $7,5 * 0,20 = 1,50$ | • Prova 1:                  | $8,0 * 0,20 = 1,60$ |
| • Prova 2:                  | $8,0 * 0,25 = 2,00$ | • Prova 2:                  | $7,0 * 0,25 = 1,75$ |
| • Trabalho Final:           | $9,0 * 0,35 = 3,15$ | • Trabalho Final:           | $7,0 * 0,35 = 2,45$ |
| • Exercícios:               | $10 * 0,20 = 2,00$  | • Exercícios:               | $0,0 * 0,20 = 0,00$ |
| • <b>Média final:</b> ..... | <b>8,65</b>         | • <b>Média final:</b> ..... | <b>5,80</b>         |

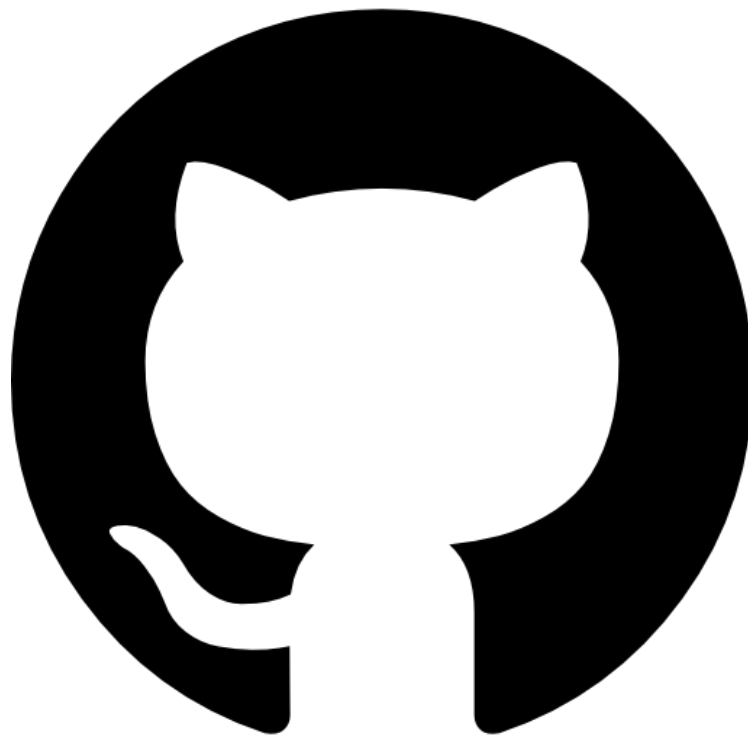


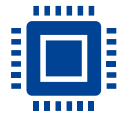
# Prova de suficiência





# Material





# Fundamentos da programação de computadores

- Computadores – Máquinas programáveis
- Ações baseadas em dados e regras
- Programador
- Algoritmos, Fluxogramas, Diagramas



# Algoritmo

Um Algoritmo é uma sequência de instruções passo a passo para resolver um problema.

- Listar os ingredientes
- Preparar esses ingredientes
- Montar o sanduíche
- Servir o sanduíche



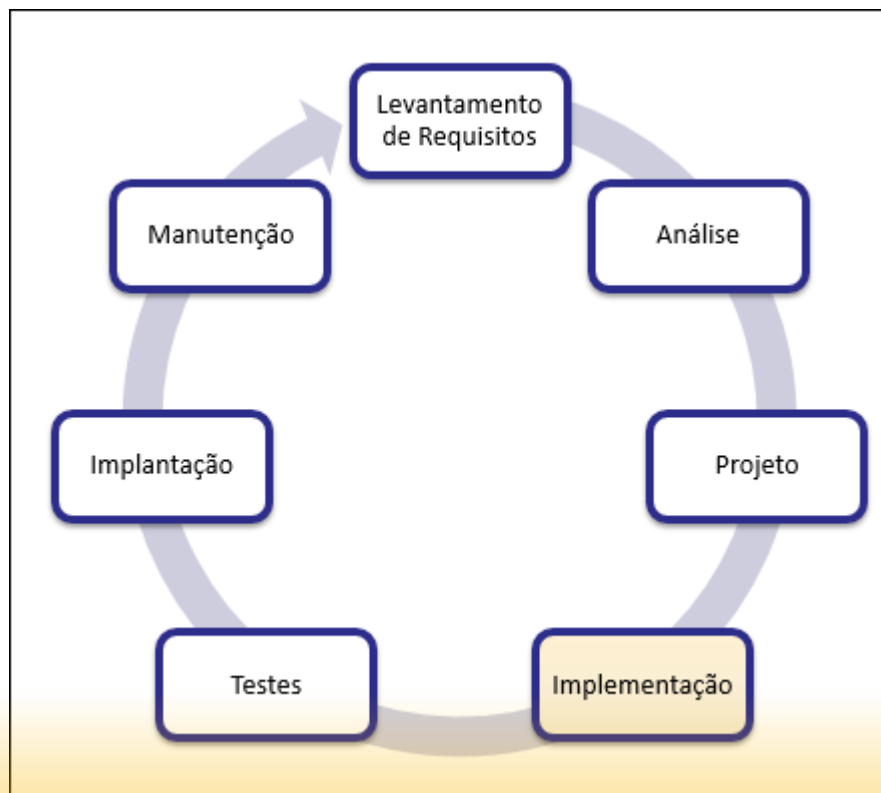


# Solução de problemas





# Ciclo de vida de um software





# Métodos de especificações de problemas

- Abordagem Estruturada

O programa é dividido em blocos de código chamados funções ou procedimentos, que realizam tarefas específicas. Cada função recebe dados como entrada, processa esses dados e pode retornar um resultado.

- Abordagem Orientada a Objetos

O programa é modelado em torno de objetos, que são instâncias de classes. Cada objeto possui características (atributos) e comportamentos (métodos) associados a ele.

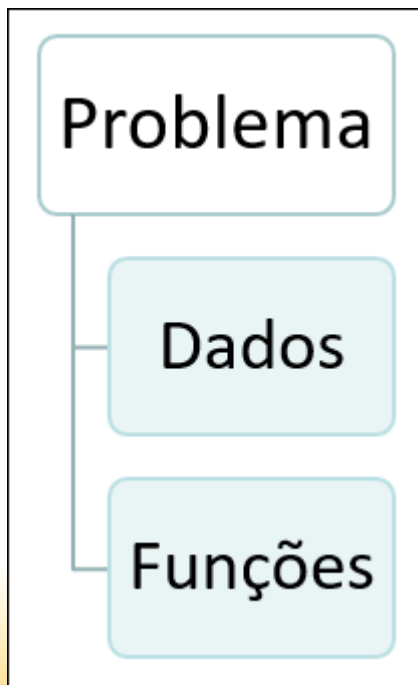
Exemplo: Calculo de média de alunos





# Métodos de especificações de problemas

- Abordagem Estruturada



- Abordagem Orientada a Objetos





# Técnicas para Representação de Soluções

- Algoritmos

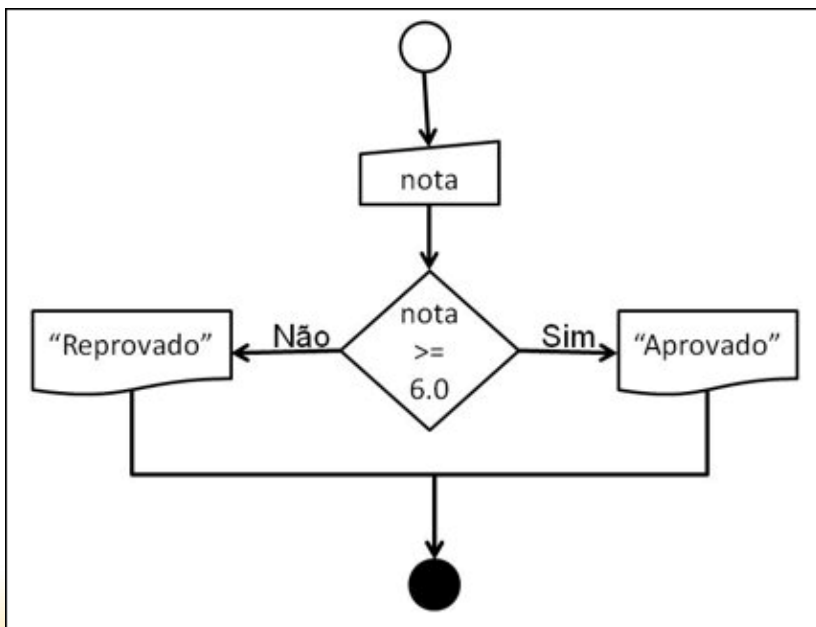
```
ler (nota)
se nota >= 6.0 então
    escrever ("Aluno aprovado")
senão
    escrever ("Aluno reprovado")
fim se
```

pseudolinguagem



# Técnicas para Representação de Soluções

- Diagramas



Fluxograma

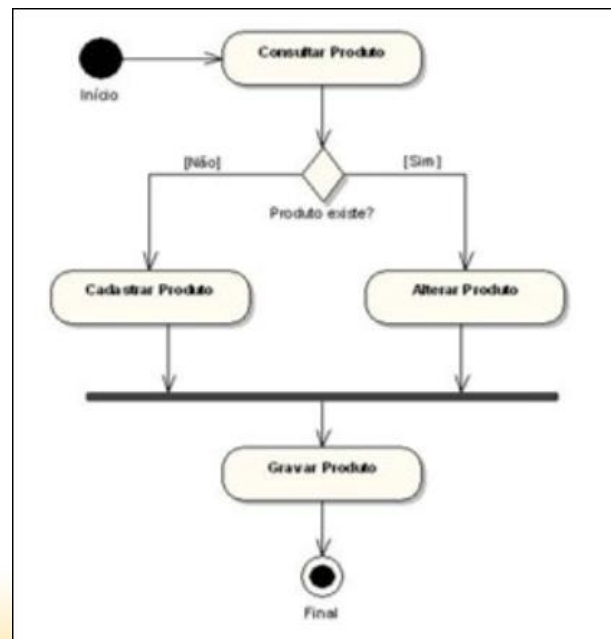


Diagrama de estados

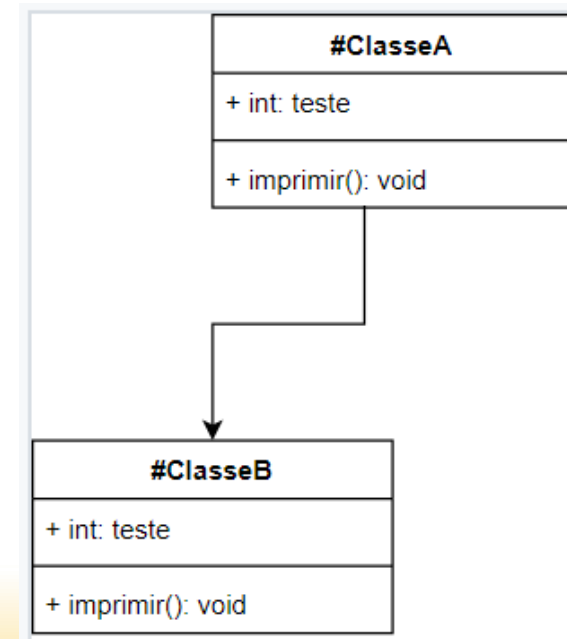
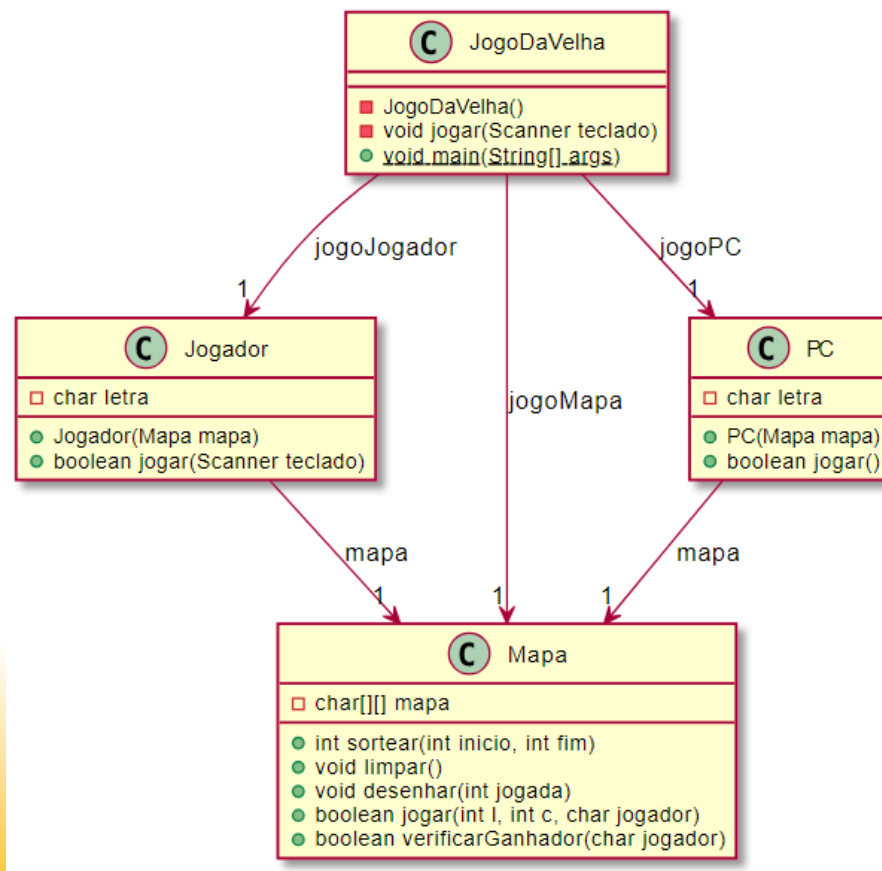


Diagrama de Classes

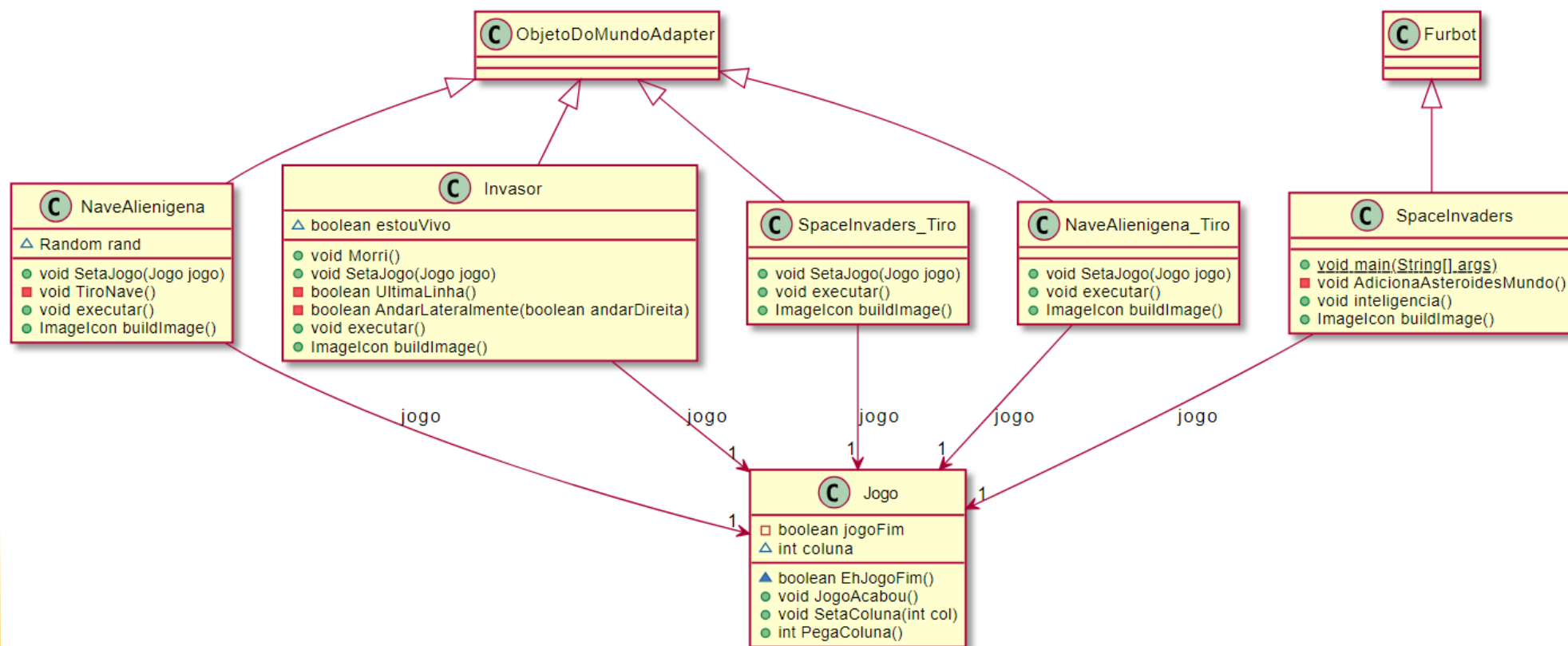


# Técnicas para Representação de Soluções





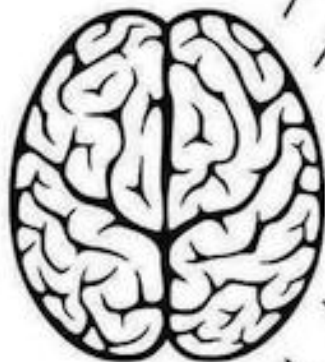
# Técnicas para Representação de Soluções



# ✓ Solução de Problemas

- A solução de um problema computacional requer
  - Busca de outras experiências;
  - Divisão e conquista;
  - Diversidade de soluções;
  - Escolha da linguagem;
- Para ser um bom programador é necessário
  - Estudar soluções algorítmicas;
  - Conhecer as estruturas e recursos das linguagens;
  - Praticar, praticar e praticar;
  - Escolha da linguagem;

## COMO NOSSO CEREBRO APRENDE?



10%



LENDO

20%



OUVINDO

30%



OBSERVANDO

50%



ASSISTINDO

70%



DEBATENDO

80%



**PRATICANDO**

95%



ENSINANDO



# Praticando a solução de problemas







# Interpretando os enunciados

- Entradas (substantivos)
- Definir os processos (ações/verbos)
- Saídas
- Dados de entrada
- Dados intermediários
- Dados de saída





# Exemplos de enunciados

Era uma vez, uma pessoa que estava na praia. Esta pessoa conhecia um professor e este professor tinha um problema para resolver. O seu problema era em determinada situação ele poderia ter 3 notas de um aluno e com elas poderia calcular a média deste aluno.



# Exemplos de enunciados

Considerando que um professor tem notas de um aluno, calcule média do aluno.



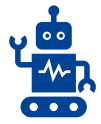
# Exemplos de enunciados

Um professor deseja calcular a média de um aluno. O professor tem o nome e as 3 notas do aluno e a média deve ser calculada de forma aritmética.

Você consegue resolver este problema sem o uso do computador?

- 3 notas;
- Calcular a média;

$$\frac{\text{Nota1} + \text{Nota2} + \text{Nota3}}{3} = \text{Média}$$



# Algoritmo - Pseudo linguagem

Sequência de instruções ordenadas de forma lógica para a resolução de uma determinada tarefa ou problema

```
ler (nota)
se nota >= 6.0 então
    escrever ("Aluno aprovado")
senão
    escrever ("Aluno reprovado")
fim
```

- Desenvolver a ideia
- **Formalizar a ideia**
- Codificar
- Testar



# Interpretando os enunciados

Um professor deseja calcular a média de um aluno. O professor tem o nome e as 3 notas do aluno e a média deve ser calculada de forma aritmética.

Entradas:

Nota1 = 7

Nota2 = 8

Nota3 = 9

Processos:

Calcular a média

$media = (nota1 + nota2 + nota3) / 3$

$media = (7 + 8 + 9) / 3$

Saída:

escreve(media)

escreve(8)



# Testes

Entradas	Saída
Nota1 = 5.0, Nota2 = 7.0, Nota3 = 9.0	Média = 7.0
Nota1 = 4.0, Nota2 = 6.0, Nota3 = 5.0	Média = 5.0
Nota1 = 7.0, Nota2 = 5.0, Nota3 = 1.5	Média = 4.5
Nota1 = 7.0, Nota2 = 8.0, Nota3 = 9.0	Média = 8.0



# Interpretando os enunciados

Modele a solução de um problema para calcular a área de uma sala retangular.  
Considere que as medidas estão expressas em metros.

Comprimento: 7 metros

Largura: 5 metros

Entradas	Saída
Comprimento: 10, largura:10	Área = 100
Comprimento: 3, largura:5	Área = 15
Comprimento: 90, largura:3	Área = 270
Comprimento: 7, largura:9	Área = 63