

# CSS

## Aula 09 Introdução à Lógica com React I



REALIZAÇÃO:

**SENAI** J.P.Morgan

# // Agenda



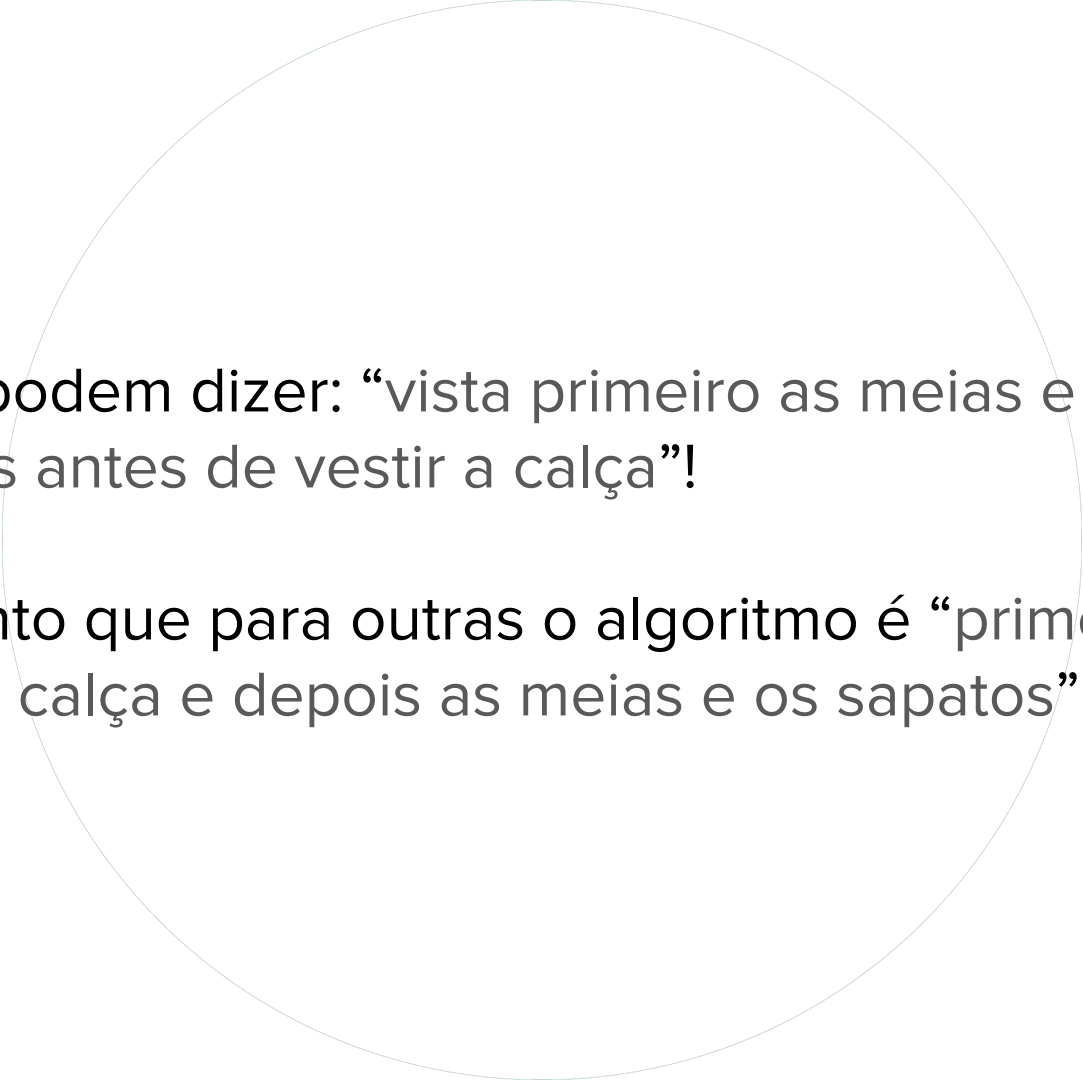
# O que é programar?

É Instruir um computador a executar uma tarefa pra gente e para isso usamos algoritmos!

**Algoritmos** são “os passos necessários para realizar uma tarefa”.

## Exercício 1:

Imaginem que a tarefa é se vestir.  
Quais são os passos para vestir uma calça,  
meias e sapatos?



Umas podem dizer: “vista primeiro as meias e os sapatos antes de vestir a calça”!

Enquanto que para outras o algoritmo é “primeiro vestir a calça e depois as meias e os sapatos”...

# Diferentes algoritmos!

**Apesar de ambos levarem ao mesmo resultado, fica claro que o primeiro algoritmo é mais difícil de executar que o segundo!**

**Quase sempre não acertamos a melhor solução de primeira, mas calma...**

# ...Refatoração

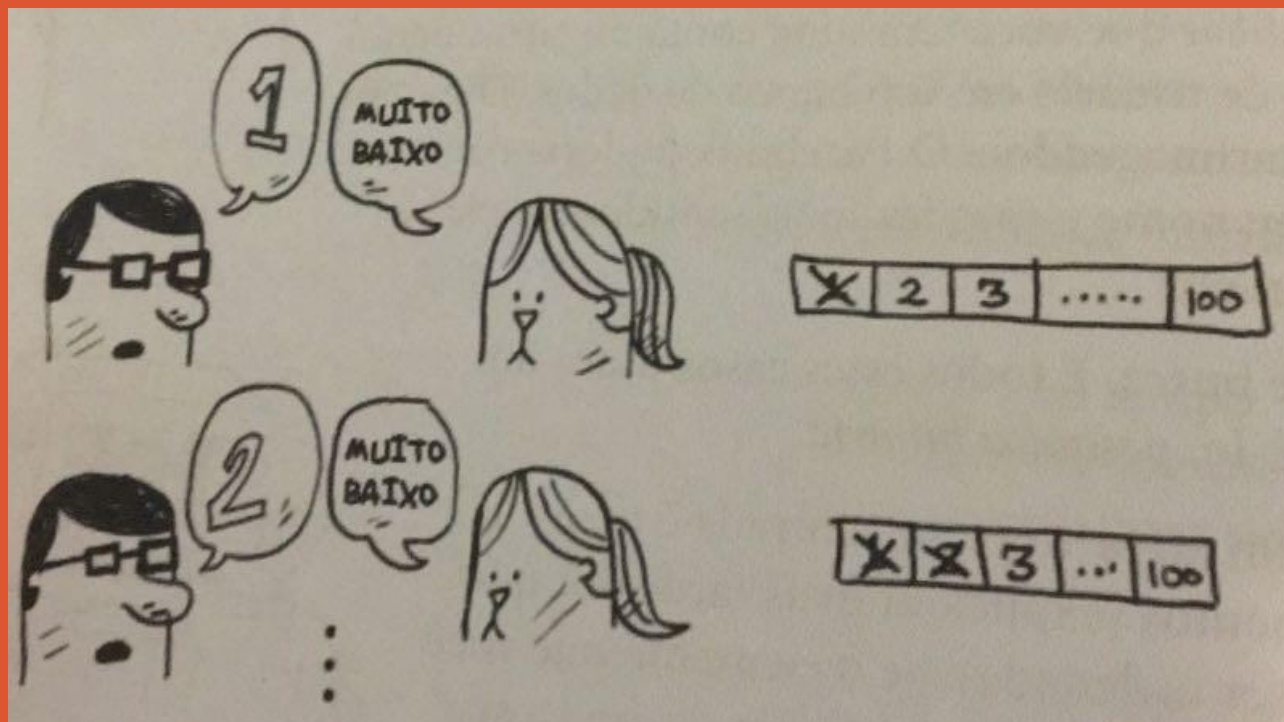
Na programação estamos sempre reescrevendo nosso algoritmo melhorando ele.

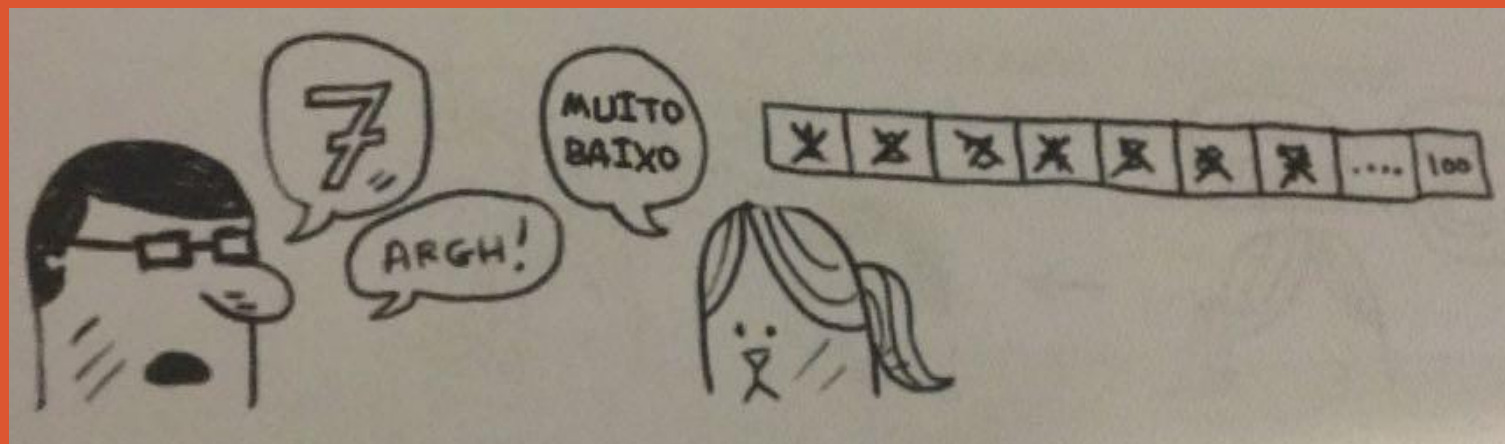
Chamamos as melhorias de refatoração.

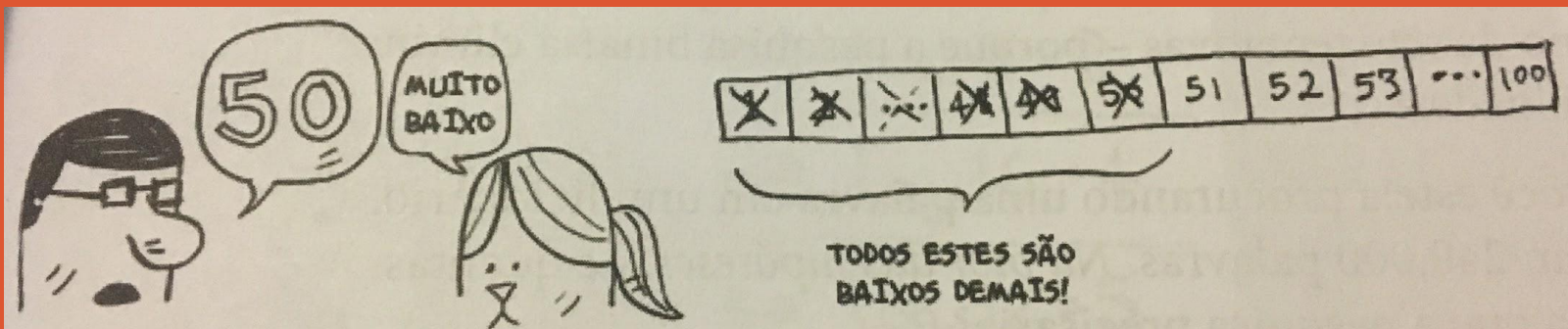
## Exercício 2:

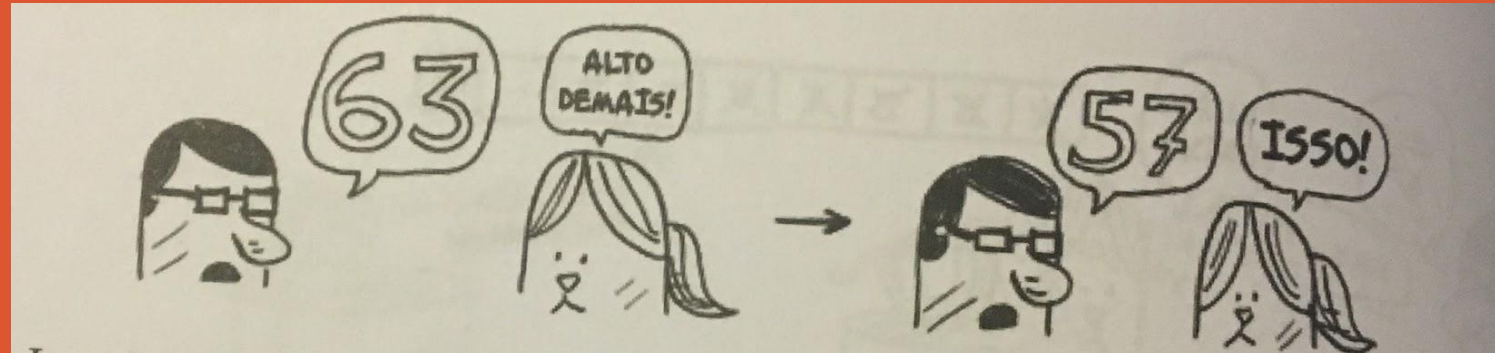
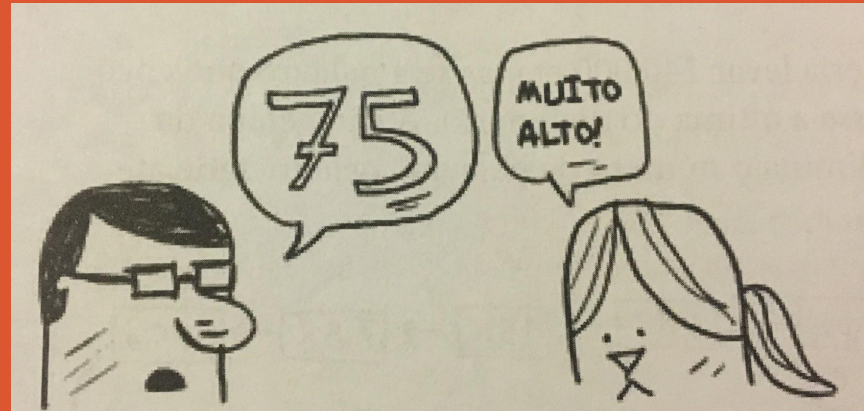
Vocês têm de adivinhar um número que eu pensar com o menor número de tentativas, e a cada tentativa digo se chutaram muito pra cima ou baixo!

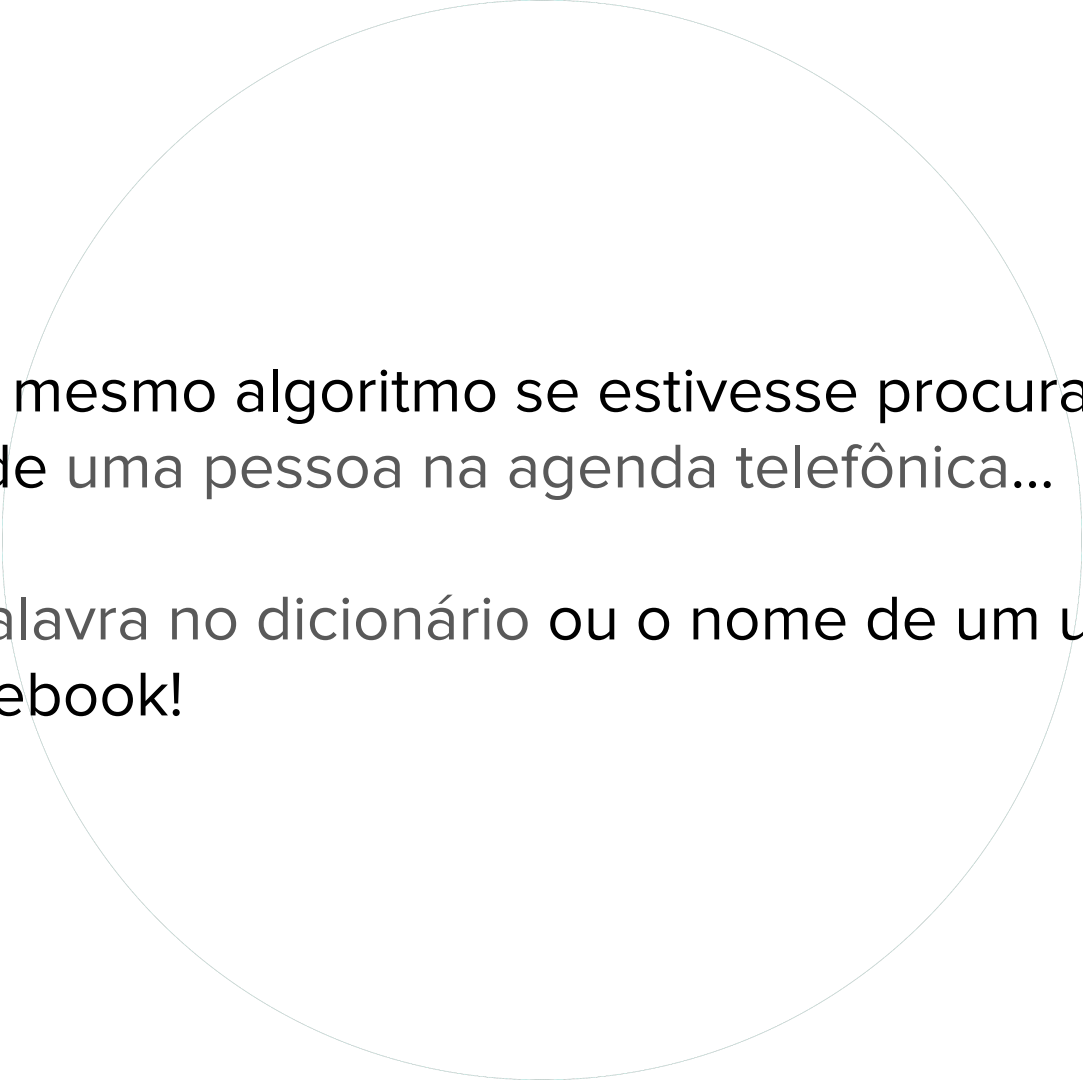












Seria o mesmo algoritmo se estivesse procurando o nome de uma pessoa na agenda telefônica...

Uma palavra no dicionário ou o nome de um usuário do Facebook!

# Antes de começarmos a programar, desenhamos um diagrama com os passos do algoritmo:

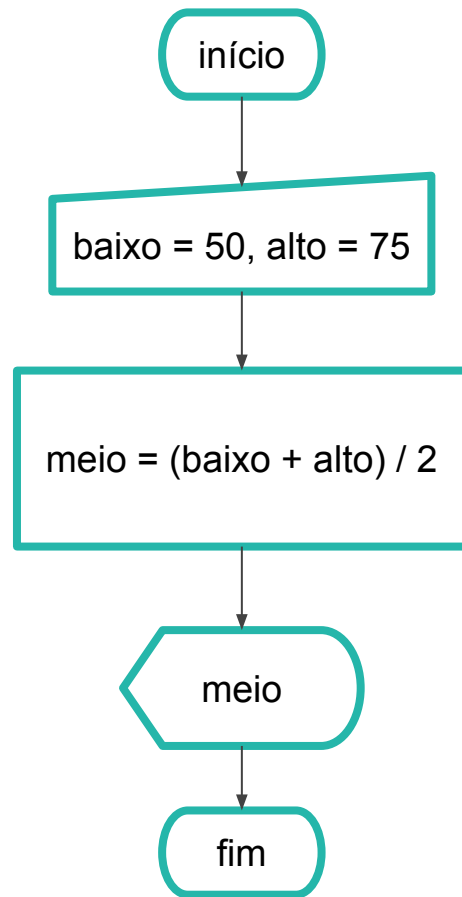
**50** é muito baixo...

**75** é muito alto...

Qual é o número **intermediário**?

$$50 + 75 = 125$$

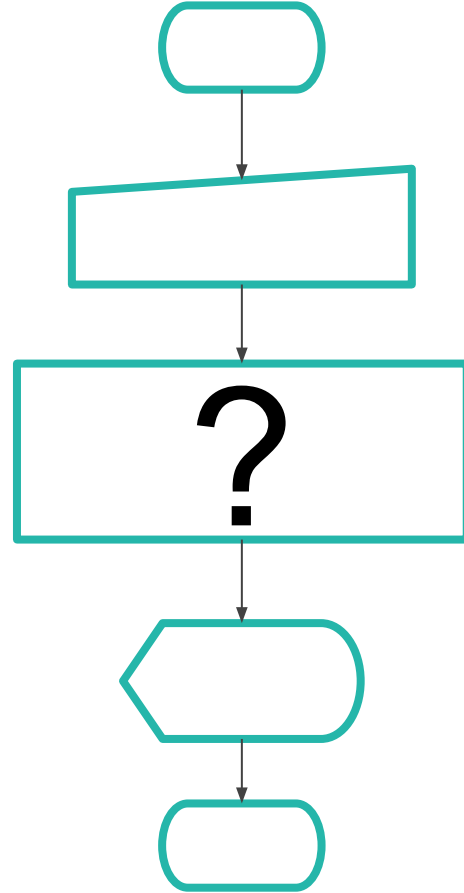
$$125 / 2 = \mathbf{63}$$



## Exercício 3, Agora é com vocês!

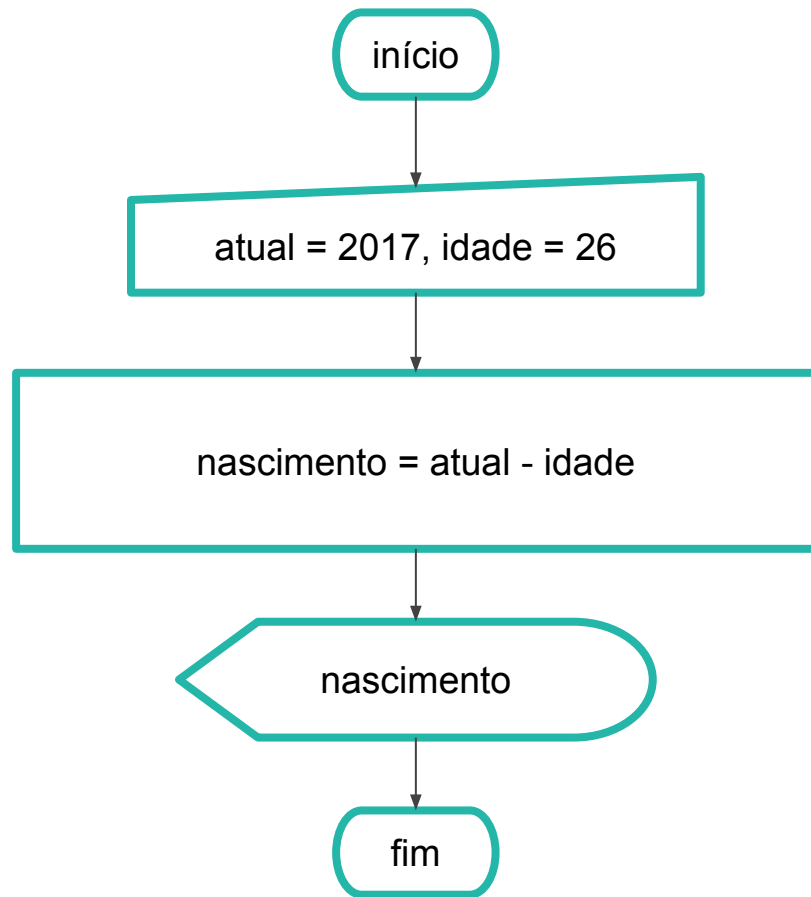
Faça um fluxograma  
do algoritmo que calcule o ano que nasci  
sabendo que tenho 27 anos:

O cálculo é: **ano atual - idade**.



# A resposta é!

2017 - 27 = 1990.

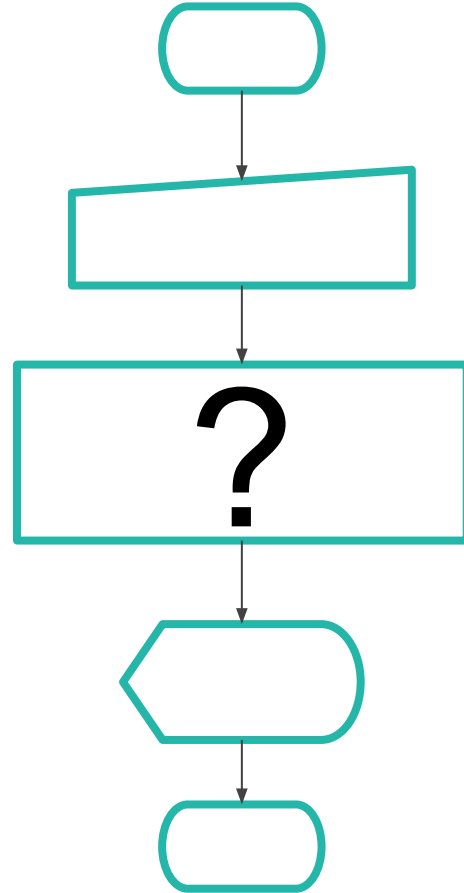




## Exercício 4, Pra ter certeza que entendeu!

Faça um fluxograma do algoritmo que calcule média final de um aluno cujas notas foram 6, 7, 8:

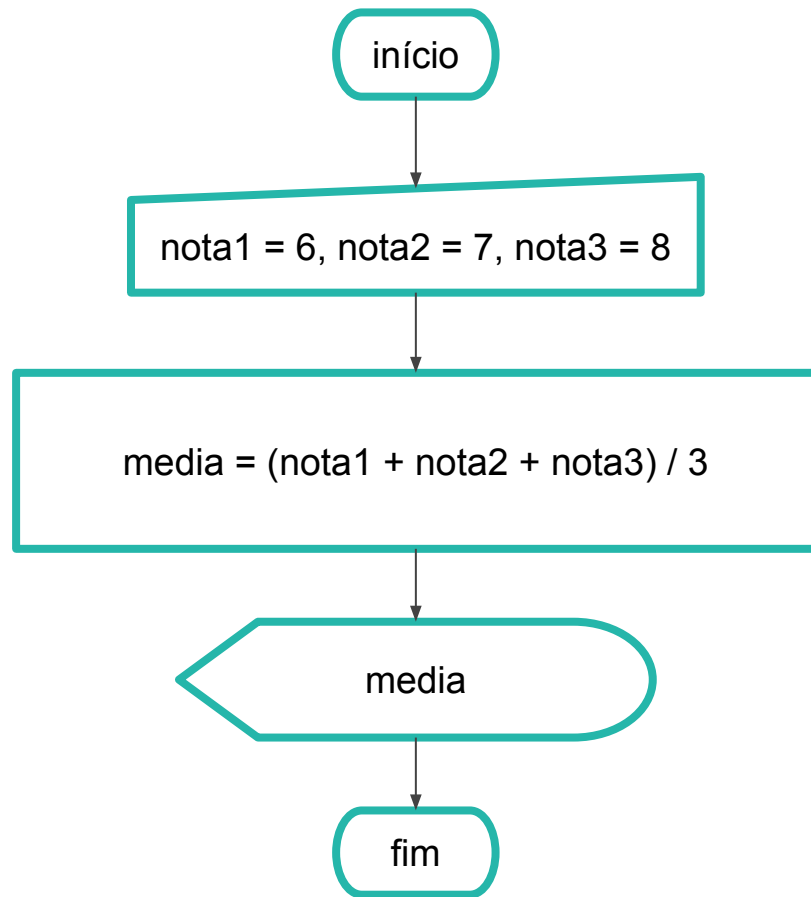
O cálculo é:  $(\text{nota1} + \text{nota2} + \text{nota3}) / 3$ .



# A resposta é!

$$6 + 7 + 8 = 21$$

$$21 / 3 = 7$$



# Entrada, Processamento e Saída

A entrada é quase sempre uma tela onde um usuário informa dados. Já processamento aquilo que fazemos com dados para gerar uma saída, dados processados



Ingredientes

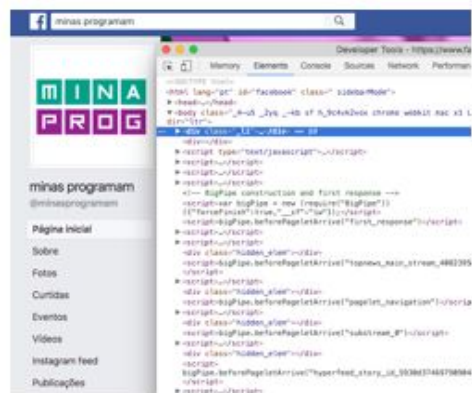


Dados de perfil

ENTRADA



Mistura de ingredientes



Lógica programada no Facebook

PROCESSAMENTO



Bolo



Sugestões de eventos, pessoas e páginas

SAÍDA

# Tipos de dados na tela

Os dados podem ser texto (costumamos chamar texto de string), à números do tipo inteiro, decimal (que no mundo da programação são mais conhecidos como double ou float) e dados de um tal de tipo booleano, esses são os tipos primitivos, mais simples...

Ah, booleano é a resposta sim ou não, por exemplo

## Publicações



minas programam

15 h ·

gente, a mina incrível do dia é a Loreнна Villas Boas.

entre os dias 14 e 19 de maio rolou a Intel ISEF (Feira Internacional de Ciência e Engenharia). na ocasião, 33 estudantes do Brasil estavam apresentando 18 projetos selecionados. uma dessas estudantes é a baiana Loreнна Villas Boas, que criou o projeto "JustStep: piso tátil integrado a comando de voz".

através de dispositivos de áudio que interagem com mecanismos de alerta em pisos internos, o mecanismo auxiliaria pessoas... Ver mais



Estudantes brasileiros arrasam na maior Feira Internacional de Ciências e Engenharia do mundo

À convite da Intel, o Hypeless foi a Los Angeles (EUA) conferir a Intel ISEF (International Science and Engineering Fair), onde 33 estudantes brasileiros...

HYPENESS.COM.BR



Curtir



Comentar



Compartilhar



45

Ordem cronológica

E caso o computador responda “**Olha, não foi curtida não**”, aí nós guardaríamos a resposta “*não é verdade!*”, que traduzida para o Inglês seria **false!**

**Temos também os chamados de complexos, um exemplo de tipo de dado complexo seria a hora que um post foi feito no Face, tipo postado às 15h (horas).**



O nome só é complexo porque na variável guardamos um tipo de dado que é uma mistura de outros tipos de dados, uma data não tem mês, dia e ano?

Dia e mês e ano seriam de que tipos?



```
Memory Elements Console
top Filter
> var data = {
    dia: 1,
    mes: "junho",
    ano: 2017
};
< undefined
> data.dia
< 1
> data.mes
< "junho"
> data.ano
< 2017
> |
```

Digamos que hoje seja dia 1 de Junho de 2017, “1” é número inteiro e “Junho” um texto, então data é um tipo de dado que é formado por um conjunto (ou complexo) de outros tipos de dados.



Mas onde ficam esses dados?



Mas onde ficam esses dados?

DUAS GAVETAS,  
POR FAVOR!

PODE USAR ESTAS DUAS  
GAVETAS, MONSIEUR



# Variáveis primitivas e objeto

Região na memória do computador onde armazenamos dados de um programa.

**Tem esse nome de variável porque** o dado armazenado pode ser alterado ao longo do tempo, **hoje pode ser camiseta, amanhã calça e depois meias...**

## Javascript:

```
Memory Elements Console
top Filter
> var nome = "Camila Belo";
< undefined
> nome
< "Camila Belo"
> var idade = 27;
< undefined
> idade
< 27
> var avaliacao = 4.5;
< undefined
> avaliacao
< 4.5
> var curtida = true;
< undefined
> curtida
< true
```

comando para criar uma variável  
(palavra reservada da linguagem)

**var** nome = "Camila Belo"

nome da  
variável  
(minúsculo, s/acentos)

dado armazenado  
na variável  
(o tipo é inferido como **texto**)

**Uma variável só pode armazenar um tipo de dado e toda vez que o computador armazenar um dado em uma dessas divisões, o dado que antes estava armazenado é eliminado...**

Por exemplo: **tenho 2 variáveis com os nomes das professoras hoje, só que me confundi, estão trocados os nome, troque-os.**

Deu ruim:

```
Memory Elements Console
top ▼ Filter
> var professora1 = "Marylly Silva";
< undefined
> var professora2 = "Camila Belo";
< undefined
> var professora1 = professora2;
< undefined
> professora1
< "Camila Belo"
> professora2
< "Camila Belo"
> // Xiii, cadê o nome da Marylly?
```

Agora sim:

```
Memory Elements Console
top ▼ Filter
> var professora1 = "Marylly Silva";
< undefined
> var professora2 = "Camila Belo";
< undefined
> var temporaria = professora1;
< undefined
> professora1 = professora2;
< "Camila Belo"
> professora2 = temporaria;
< "Marylly Silva"
> professora1
< "Camila Belo"
> professora2
< "Marylly Silva"
>
```



Uma variável só pode guardar um dado!  
Se armazenamos um dado em uma variável que já tinha outro armazenado, este último será sobrescrito.



Em Javascript nós costumamos chamar os tipos de **dados complexos** de **objetos**.

Fazemos isso porque podemos criar variáveis **para representar os objetos do mundo real**, em um sistema!

### Exercício 5:

Imagine que você trabalha no facebook, crie uma **variável** para armazenar o nome, email, senha e telefone de uma **pessoa**.

EX. Camila, camila@email.com, 123456, 11 99999-999

E pra finalizar, agora você mudou de emprego e trabalha numa concessionária de carros, criei uma **variável** que tenha um objeto para armazenar o modelo, ano, marca, preço e cor de um **carro**.

EX. March, 2017, Nissan, Rosa



```
Memory Elements Console
top Filter
> var pessoa = {
  nome: "Camila",
  email: "camila@email.com",
  senha: "Abc1234",
  telefone: "11981598829"
};
< undefined
> pessoa.nome
< "Camila"
> pessoa.email
< "camila@email.com"
> pessoa.senha
< "Abc1234"
> pessoa.telefone
< "11981598829"
> |
```

Os dados estão entre chaves e separados por vírgula.  
Todo comando de criação de variável deve terminar com ponto e vírgula!

```
Memory Elements Console
top Filter
> var carro = {
  modelo: "March",
  ano: 2017,
  marca: "Nissan",
  cor: "Vermelho",
  preco: 40.000
};
< undefined
> carro.modelo
< "March"
> carro.ano
< 2017
> carro.marca
< "Nissan"
> carro.cor
< "Vermelho"
> carro.preco
< 40
> |
```

Textos ficam entre aspas duplas, valores com casa decimal, como o preço do carro, por exemplo, devem usar ponto ao invés de vírgula!

[ Escola SENAI de informática ]

Alameda Barão de Limeira, 539, Santa Cecília - São Paulo - SP

[informatica.sp.senai.br](http://informatica.sp.senai.br)

