

Algoritmo

Vamos fazer um algoritmo para fazer um bolo

- Os ingredientes são definidos antes;
- Modo de preparo depois;



Algoritmo

```
FazerBolo(){  
    //Ingredientes  
    Acucar = 2 xícaras;  
    Farinha = 2,5 xícaras;  
    Ovos = 4;  
    Oleo = 0,5 xícaras;  
    Fermento = 1 colher;  
    FormaDeBolo = nada;  
  
    //Modo de preparo  
    //Misturar os ingredientes “até homogeneizar”  
    formaDeBolo = Acucar + Farinha + Ovos + Oleo + Fermento ;  
    ColocarFormaNoForno(formaDeBolo);  
    Aguardar assar por 1 hora  
}
```

O que é um Programa?

O que é um Programa?

É um algoritmo escrito em uma linguagem de programação;



Portugol Studio

- Foco é entender os princípios da programação.
- Possui uma linguagem própria
- Aproxima a linguagem de programação ao português

Portugol Studio – Estrutura do código.

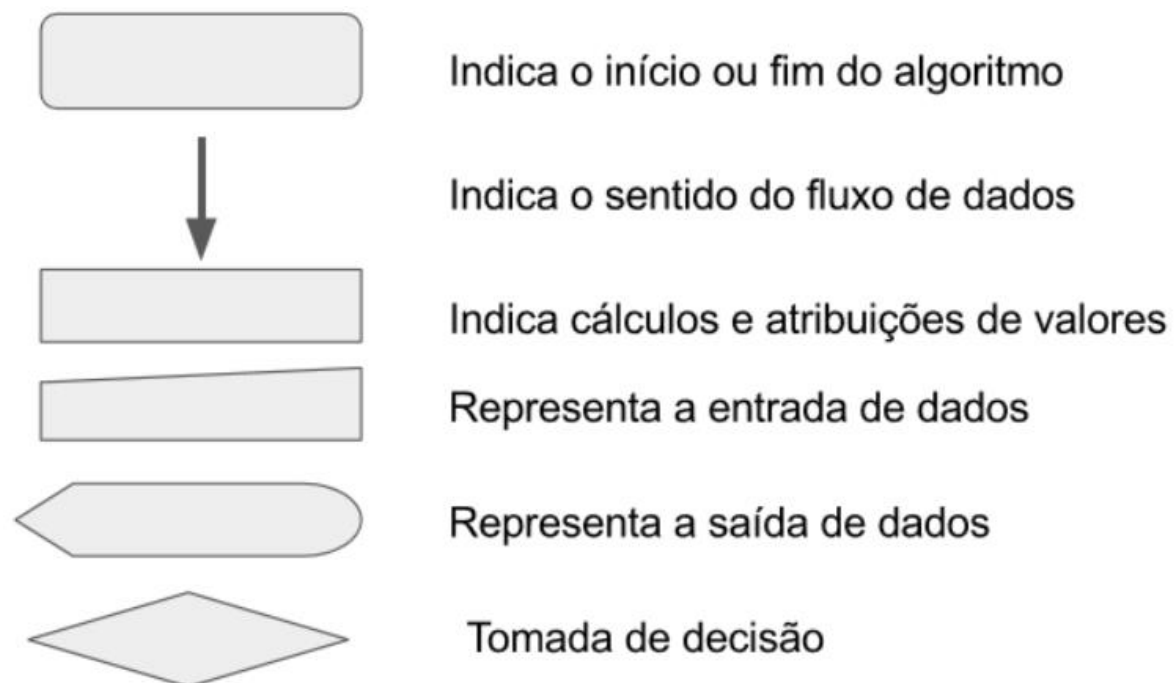
Pseudo Código?

Programa

```
{  
    /* Declaração de variáveis, estruturas e outras funções */  
    funcao inicio ()  
    {  
        /*Execução da função início*/  
    }  
}
```

Fluxograma

- É uma forma universal de representação, pois se utiliza de figuras geométricas para ilustrar passos a serem seguidos



Revisando...

Já aprendemos:

- Valores
 - Características que devemos adquirir
 - As regras do jogo
 - A importância do nivelamento
 - Organizar o pensamento
- Conteúdo
 - O que é um algoritmo
 - O que é um programa
 - Qual ferramenta utilizaremos
 - Representações de algoritmos
 - Pseudocódigo
 - Fluxograma

Vamos instalar o Portugol Studio

<http://lite.acad.univali.br/portugol/>

Primeiro Programa

Nosso primeiro programa: Olá mundo!

Execute no Portugol Studio o código : Olá Mundo

- O que esse código faz?
- Quais dificuldades vocês tiveram em entender este trecho de código?

Outros Programa

Identificando-se : Qual é o seu nome?

Execute os seguintes programas - **‘Número Digitado’** e **‘Seu Nome’**

- O que esses códigos fazem?
- A partir deles, escreva um novo programa que recebe seu nome e escreve o seu nome na tela

Operações de Entrada e Saída

Ficou parecido com a solução abaixo?

programa

```
{  
    funcao inicio ()  
    {  
        cadeia nome /*cadeia se refere ao tipo da variável que é uma cadeia de caracteres*/  
  
        escreva("Digite seu nome: ")  
        leia(nome)  
        escreva("Seu nome é : ", nome , "\n")  
    }  
}
```

Falaremos depois
sobre **variáveis** e
seus **tipos** !!

Por que entrada e saída?

Quando escrevemos :

cadeia nome

leia(nome)

- **leia** é uma operação de **entrada**;
- Permite que o que escrevemos no teclado seja **lido e armazenado** na variável “nome”;
- Logo estamos **entrando** com uma informação no programa durante sua execução;

Por que entrada e saída?

Quando escrevemos :

cadeia nome = Fulano

escreva("Meu nome é: ", nome)

- **escreva** é uma operação de **saída**;
- **P**ermite que a informação escrita entre seus parênteses () seja apresentado na tela do computador;
- Logo como é uma informação de apresentação, entendemos como uma informação de **saída**;

Variáveis e constantes

Voltando ao programa anterior...

```
programa
```

```
{
```

```
    funcao inicio ()
```

```
{
```

```
    cadeia nome /*cadeia se refere ao tipo da variável que é uma cadeia de caracteres*/
```

```
    escreva("Digite seu nome: ")
```

```
    leia(nome)
```

```
    escreva("Seu nome é : ", nome, "\n")
```

```
}
```

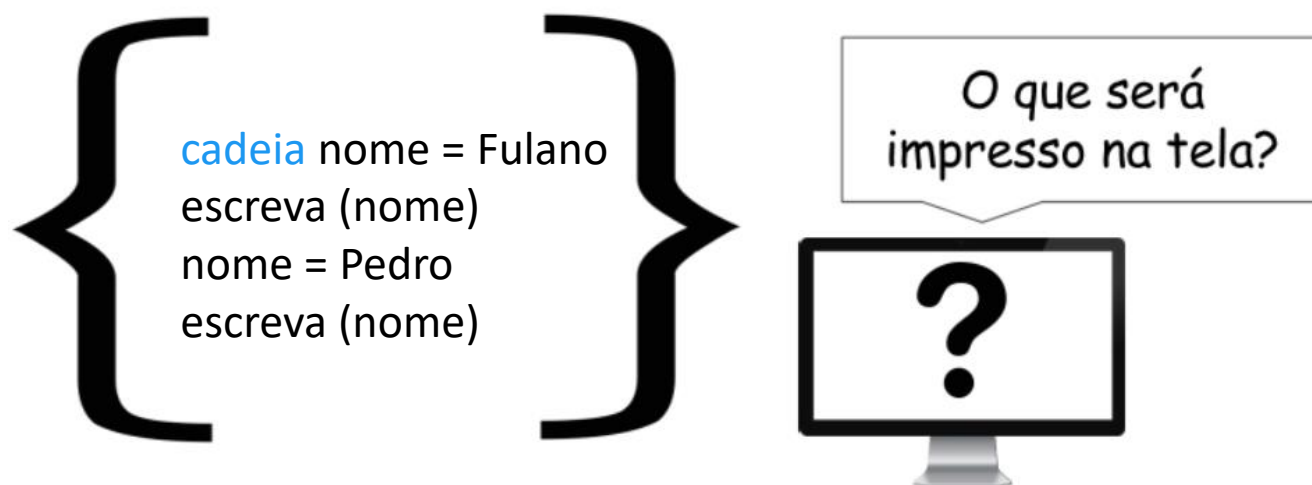
```
}
```

Ficou faltando entendermos o que são as **variáveis**!



Variáveis e constantes

- **Variáveis e constantes** representam uma posição na memória, onde pode ser armazenado um **único dado** (valor);
- Possuem um **nome** e um **valor**;
- A diferença entre variáveis e constantes na execução do programa:
 - Valor da variável pode mudar
 - Valor da constante não.

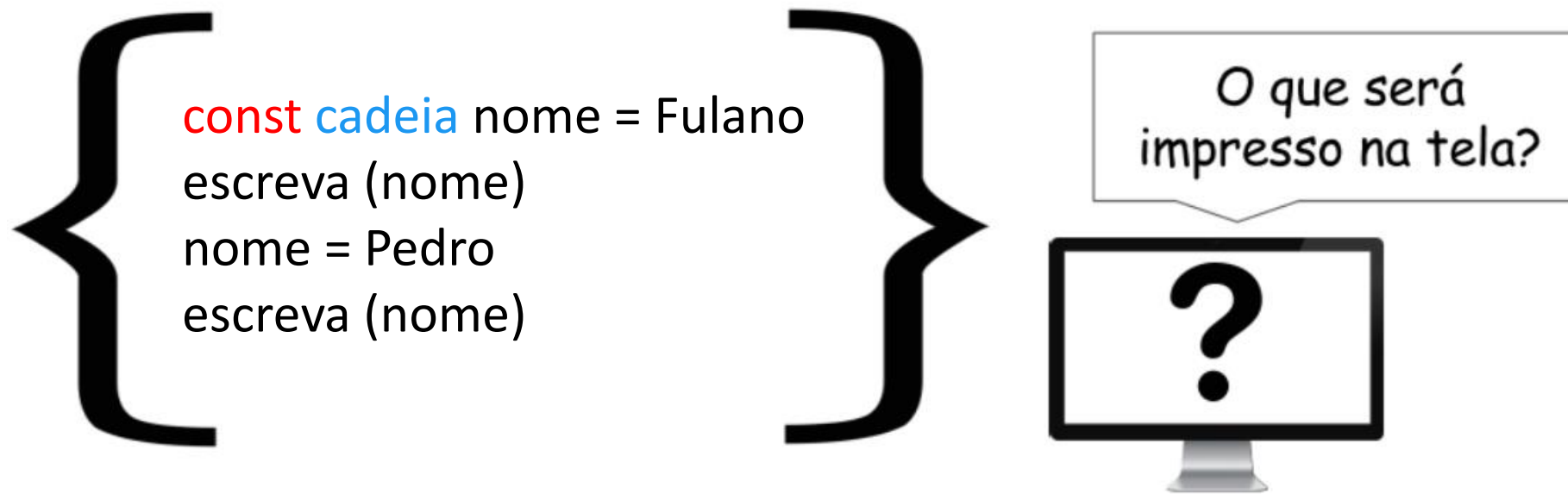


Variáveis e constantes

- Em algumas linguagens (incluindo Portugol) as variáveis podem ser tipadas, ou seja, aceitam apenas valores referentes ao seu tipo, representado antes do nome da variável:
- o Tipos na linguagem do Portugol Studio
 - inteiro : Número inteiros -> 1 ; 2 ; 3
 - real : Números de ponto flutuante -> 1.1 ; 3.14 ; 10.3
 - cadeia : Cadeia de caracteres -> “Adoro estudar programação”
 - character : Apenas um caractere -> “A” ou “1”
 - logico : Caractere booleano : verdadeiro, falso

Variáveis e constantes

- Finalmente, para declarar uma constante basta colocar o indicador constantes da declaração da constante;



Variáveis e constantes

- Finalmente, para declarar uma constante basta colocar o indicador constantes da declaração da constante;

const cadeia nome = Fulano
escreva (nome)
nome = Pedro
escreva (nome)

O que será
impresso na tela?



ERRO!! Pois não podemos
alterar o valor de uma
constante =(

Recapitulando...

Já aprendemos:

- O que é um algoritmo;
- O que é um programa;
- Qual ferramenta utilizaremos;
- Operações de entrada e saída;
- O que são variáveis e constantes.

O que mais precisamos aprender

- Desvios condicionais (se e senão);
- Operadores lógicos (E, OU ...);
- Laços de repetição (enquanto);
- Estruturas de dados (Vetores, Matrizes, Filas e Pilhas);
- Subrotinas (Funções);
 - Recursividade;
 - Bibliotecas.

Vamos praticar

Lembram do exemplo de escrever e imprimir?

```
programa
{
    funcao inicio ()
    {
        cadeia nome
        escreva("Digite seu nome: ")
        leia(nome)
        escreva("Seu nome é : ", nome , "\n")
    }
}
```

Vamos transformar este programa para ser capaz de **validar** o nome do usuário e conceder acesso ao sistema!
Apenas o usuário cadastrado poderá entrar no sistema.

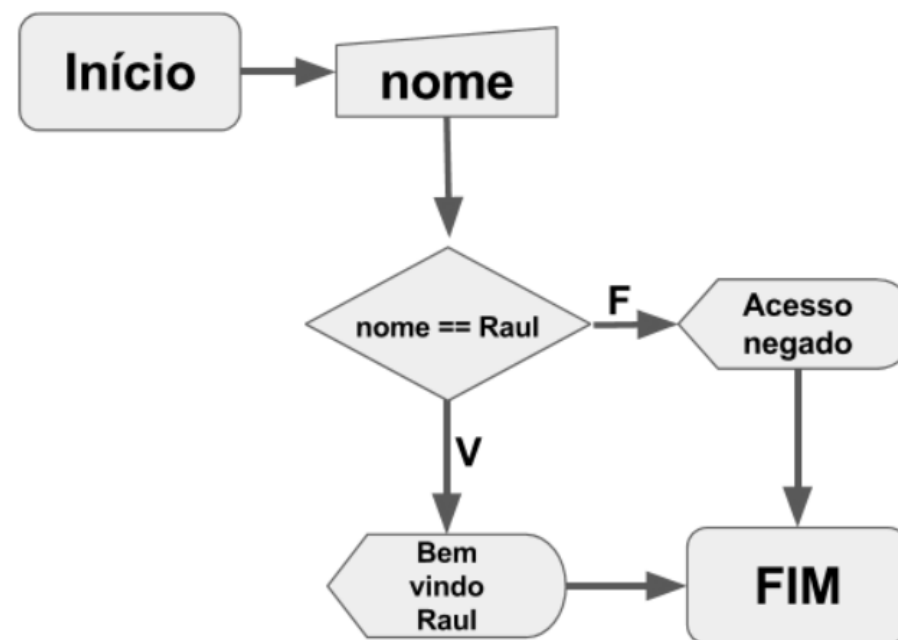
Vamos praticar

Lembram do exemplo de escrever e imprimir?

Para resolver este problema podemos usar os condicionais **se** e **senão** (if e else do inglês).

programa

```
{  
  funcao inicio () {  
    cadeia nome  
    escreva("Digite seu nome de usuário: ")  
    leia(nome)  
    se(nome == "Raul") {  
      escreva("Bem vindo ", nome, "\n")  
    }  
    senao {  
      escreva("Acesso negado!!! \n")  
    }  
  }  
}
```



Vamos praticar

- Se... então ... senão
- Como vimos, podemos utilizar as cláusulas se e senão para direcionar a execução de nosso código. A estrutura consiste em basicamente :

```
se (condição) {  
    // Execute uma parte de código  
}  
senao {  
    // Execute outra parte de código  
}
```


Vamos praticar

Será que apenas o nome de usuário é suficiente?

- Para validarmos corretamente um usuário precisamos também verificarmos se sua senha está correta. Assim precisamos validar o **nome** de usuário **E** sua **senha**.

programa

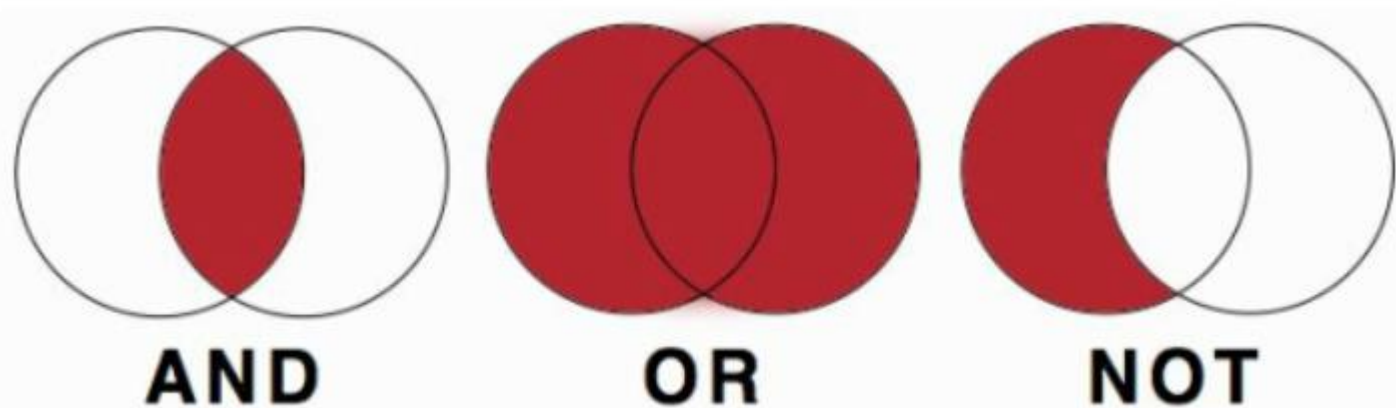
```
{
    funcao inicio () {
        cadeia nome
        escreva("Digite seu nome de usuário: ")
        leia(nome)
        se(nome == "Raul" E senha == "MinhaSenha") { /*Note o operador lógico E para verificar o usuário E senha*/
            escreva("Bem vindo ", nome, "\n")
        }
        senao {
            escreva("Acesso negado!!! \n")
        }
    }
}
```

Operadores Lógicos

- Operadores Lógicos;
- Podemos usar os operadores lógicos **E** , **OU** e **NÃO (!)** para melhorar ainda mais nossas condições;
 - Verdadeiro **E** Verdadeiro = Verdadeiro;
 - Verdadeiro **E** Falso = Falso;
 - Falso **E** Falso = Falso;
 - Verdadeiro **OU** Falso = Verdadeiro;
 - Falso **OU** Falso = Falso;
 - !Verdadeiro = Falso;
 - !Falso = Verdadeiro.

Operadores Lógicos

- Na maioria das linguagens os operadores E , OU, e NÃO são representados por `&&`, `||` e `!` , respectivamente:
 - E == `&&` == AND;
 - OU == `||` == OR;
 - NAO == `!` == NOT;



Operadores Lógicos

- A negação (!) pode ser utilizada na comparação de igual para negar uma igualdade;
- Exemplo:
 - `1 == 1` (um igual a 1)
 - `1 != 2` (um não igual a 2 || um diferente de 2)
- Outro exemplo :

```
se(nome != "Fulano") { /**/  
    escreva("Você não é o Fulano \n")  
}  
senao {  
    escreva("Olá Fulano!!! \n")  
}
```

Retrospectiva

Já aprendemos:

- O que é um algoritmo;
- O que é um programa;
- Qual ferramenta utilizaremos;
- Operações de entrada e saída;
- O que são variáveis e constantes;
- Desvios condicionais (se e senão);
- Operadores lógicos (E, OU ...).