

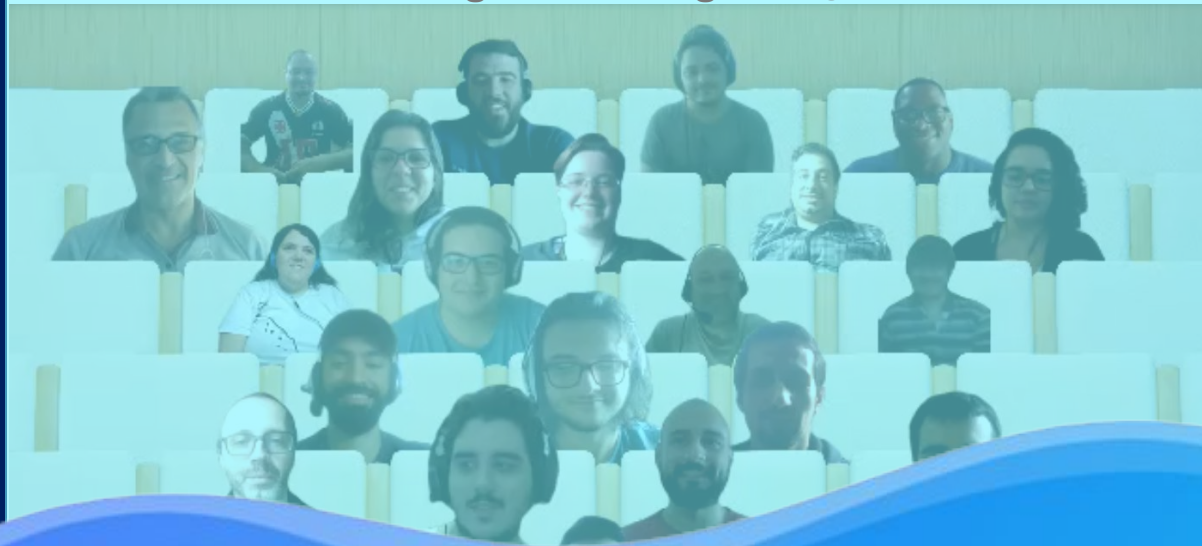
53 45 52 45 49 20 46 49 45 4c 20
41 4f 53 20 50 52 45 43 45 49 54
4f 53 20 44 41 20 48 4f 4e 52 41
20 45 20 44 41 20 43 49 c3 8a 4e
43 49 41 2c 20 50 52 4f 4d 4f 56
45 4e 44 4f 20 4f 20 55 53 4f 20
45 20 4f 20 44 45 53 45 4e 56 4f
4c 56 49 4d 45 4e 54 4f 20 44 41
20 49 4e 46 4f 52 4d c3 81 54 49
43 41 20 45 4d 20 42 45 4e 45 46
c3 8d 43 49 4f 20 44 4f 20 43 49
44 41 44 c3 83 4f 20 45 20 44 41
20 53 4f 43 49 45 44 41 44 45 2e

RESIDÊNCIA DE SOFTWARE

CAPACITAR
TREINAR
EMPREGAR
TRANSFORMAR



Lógica de Programação



Aula 2
Roni Schanuel
09-08-2022

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

**Com menos teoria e mais prática pois programar
é treinar bastante!! =D**



Lembram do exemplo de escrever e imprimir?

programa

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

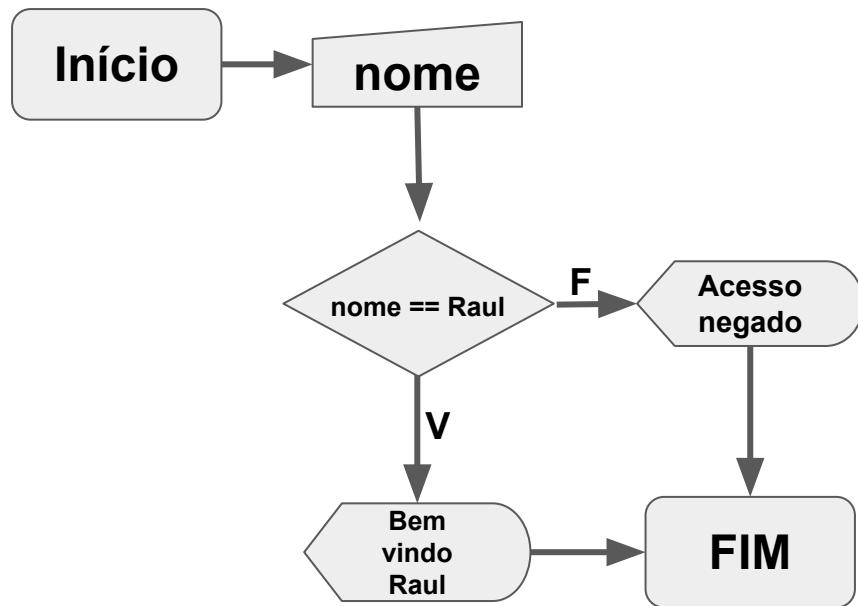
Vamos transformar este programa para ser capaz de **validar** o nome do usuário e conceder acesso ao sistema!
Apenas o usuário cadastrado poderá entrar no sistema.

Lembram do exemplo de escrever e imprimir?

Para resolver este problema podemos usar os condicionais **se** e **senao** (**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")  
        }  
    }  
}
```



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  
}
```



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

Exemplos

Operadores

Maior >

Menor <

Maior ou Igual >=

Menor ou Igual <=

Igual ==

Diferente !=

20 > 50 = F

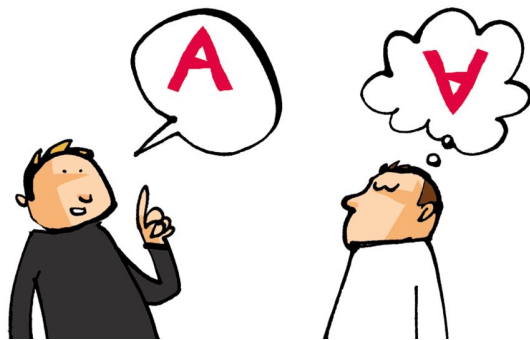
10 == 10 = V

11 != 12 = V

15 >= 10 = F

Operadores Lógicos

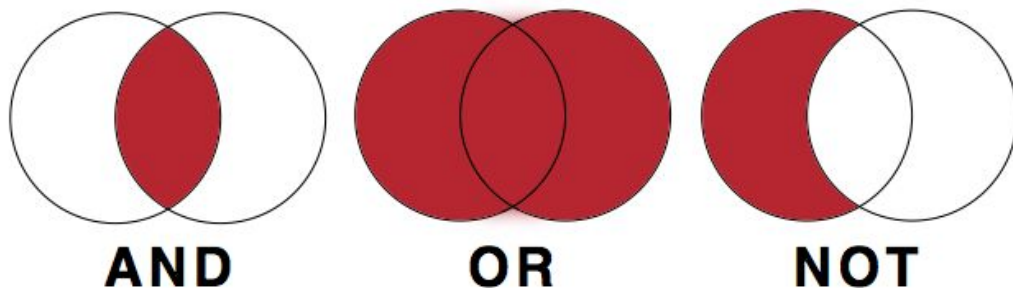
- Podemos usar os operadores lógicos E , OU e NÃO (!) para melhorar ainda mais nossas condições.
 - Entendo melhor os resultados dos operadores lógicos:
 - Verdadeiro **E** Verdadeiro = Verdadeiro
 - Verdadeiro **E** Falso = Falso
 - Falso **E** Falso = Falso
 - Verdadeiro **OU** Falso = Verdadeiro
 - Falso **OU** Falso = Falso
 - !Verdadeiro = Falso
 - !Falso = Verdadeiro
 - == (igual)
 - != (diferente, ou seja **não** igual)



Entendido?

Mais sobre operadores lógicos

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



Operador E

No operador e todas as condições tem que ser verdadeiras para o resultado ser verdadeiro

se(media ≥ 7 e faltas < 10)

A	B	Resultado
F	F	F
V	F	F
F	V	F
V	V	V

se(idade > 8 e peso > 25)

Operador OU

No operador OU pelo menos umas das condições tem que ser verdadeiras para o resultado ser verdadeiro

se(idade \geq 18 ou altura $>$ 1.85)

se(condicao == “Gestante” ou condi cao == “Deficiente”)

A	B	Resultado
F	F	F
V	F	V
F	V	V
V	V	V

Operador NAO

Este operador inverte o resultado de uma condição.

NAO (2>6) = **V**

NAO (8>10) E (6<10) = **V**
F V

Um pouco mais sobre 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")  
}
```



Caso (Condicional)

- Utilizar somente inteiros e caracteres
- Não pode utilizar operadores lógicos
- Só utilizar o operador relacional de igualdade

Escolha (variável)

caso 1

//instruções

pare

caso 2

//instruções

pare

caso 3

//instruções

pare

caso contrário

//instruções

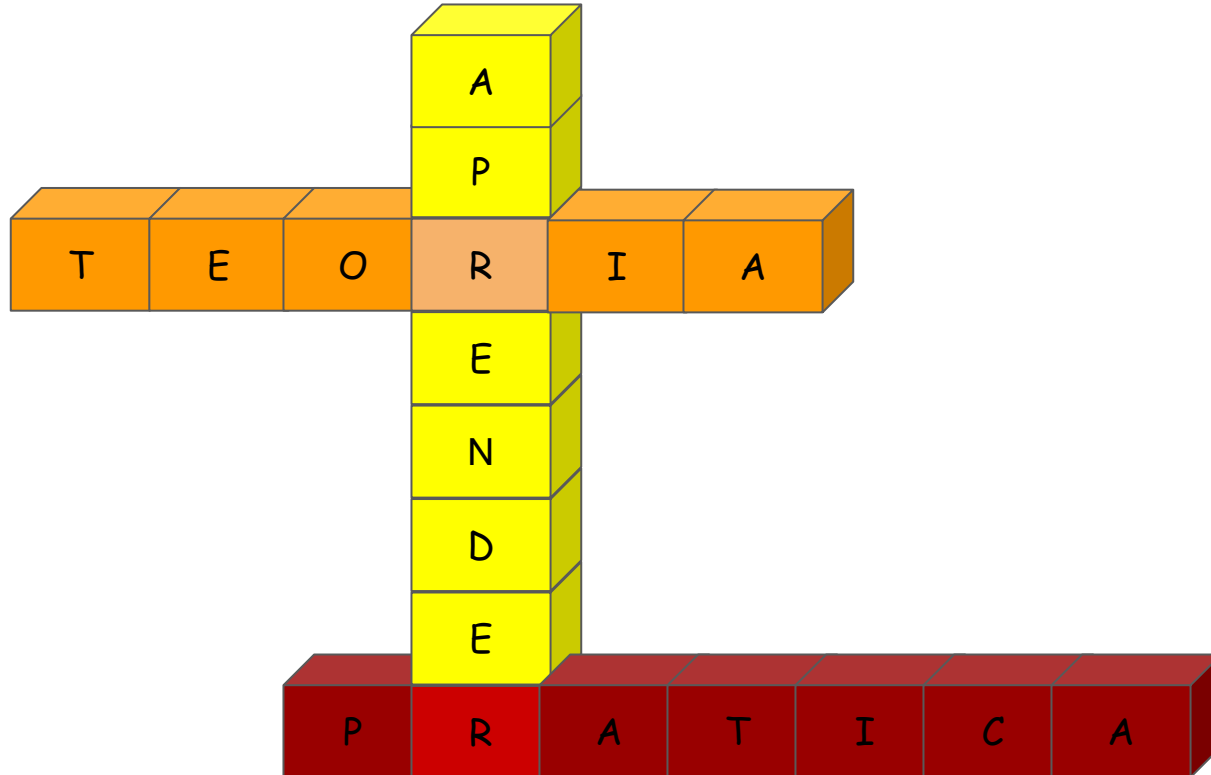
```
funcao inicio()
{
    inteiro diaSemana

    escreva("Digite o número do dia da semana (1-7)")
    leia(diaSemana)

    escolha(diaSemana){
        caso 1:
            escreva("Domingo")
            pare

        caso 2:
            escreva("Segunda-Feira")
            pare
        caso 3:
            escreva("Terça-Feira")
            pare
        caso 4:
            escreva("Quarta-Feira")
            pare
        caso 5:
            escreva("Quinta-Feira")
            pare
        caso 6:
            escreva("Sexta-Feira")
            pare
        caso 7:
            escreva("Sábado")
            pare
        caso contrario:
            escreva("Dia inválido")
            pare
    }
}
```

Praticar é fundamental



Exercícios

- 1) Leia um número e retorne como resposta se ele é positivo, negativo ou zero.
- 2) Criar um algoritmo que receba quatro notas e calcule a média. Se a média for maior que 7 deverá ser exibida a mensagem aprovado caso contrário deverá ser exibida a mensagem reprovado.

- 3) Escreva um programa que encontre o valor máximo entre 2 números

Exemplo:

Entrada:

Digite um número: 2

Digite outro número: 1

Saída:

O número 2 é maior que o número 1

- 4) Escreva um programa que funcione como uma calculadora simples de soma (+), subtração(-), multiplicação(*) e divisão(/)

Exemplo:

Entrada: 10 * 2

Saída esperada: 10 * 2 = 20

- 5) Implemente um programa que considera as seguintes opções para determinar se o usuário usará a fila preferencial ou a fila comum. O usuário usa a fila preferencial caso :

- Possui mais de 65 anos : Usa fila preferencial
- É deficiente físico : Usa fila preferencial
- É gestante : Usa fila preferencial

O programa recebe como entrada nome, a idade, e a condição especial do usuário, se houver.

Exemplo de entrada: Maria, 22, deficiente

Saída esperada: Fila preferencial

6) Faça um programa para que leia a idade e o nome de um jogador de futebol.

Categorias:

De 10-17: categorias de base

18-40: profissional

acima de 40: master

abaixo de 10: escolinha

A resposta deverá ser conforme exemplo abaixo:

Entrada:

nome: João

idade: 30

Categoria: Profissional

7) A padaria Hotpão vende uma certa quantidade de pães franceses e uma quantidade de broas a cada dia. Cada pãozinho custa R\$ 0,50 e a broa custa R\$ 5,00. Ao final do dia, o dono quer saber quanto arrecadou com a venda dos pães e broas (juntos), e quanto deve guardar numa conta de poupança (10% do total arrecadado). Você foi contratado para fazer os cálculos para o dono. Com base nestes fatos, faça um algoritmo para ler as quantidades de pães e de broas, e depois calcular os dados solicitados.

8) Um motorista deseja colocar no seu tanque X reais de gasolina. Escreva um algoritmo para ler o preço do litro da gasolina e o valor do pagamento, e exibir quantos litros ele conseguiu colocar no tanque.

9) Faça um algoritmo que leia um número e retorne como resposta se ele é par ou impar

10) Calcule o IMC conforme tabela e fórmula abaixo:

$$\text{IMC} = \frac{\text{PESO}}{(\text{ALTURA})^2}$$

IMC	Classificações
Menor do que 18,5	Abaixo do peso normal
18,5 - 24,9	Peso normal
25,0 - 29,9	Excesso de peso
30,0 - 34,9	Obesidade classe I
35,0 - 39,9	Obesidade classe II
Maior ou igual a 40,0	Obesidade classe III

Classificação segundo a OMS a partir do IMC

11) Faça um algoritmo que leia a idade de uma pessoa e de acordo com a idade exiba a seguintes mensagens:

Menor que 16 anos - não pode votar

Entre 16 e 18 anos e maior que 70 anos - voto opcional

Entre 18 e 70 anos - voto obrigatório

12) Criar um algoritmo para ler dois números do tipo inteiro e mostrar se um é múltiplo do outro em qualquer ordem de digitação.

Retrospectiva : O que já aprendemos?

- Até aqui, já vimos
 - Valores
 - Como nos organizamos como turma
 - Valores em trabalho em equipe e desenvolvimento de software
 - Conteúdo
 - 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 ...)



Retrospectiva : O que iremos aprender?

- Laços de repetição (enquanto, para ... faça)
- Subrotinas (Funções)
 - Recursividade
 - Bibliotecas
- Estruturas de dados (Vetores, Matrizes, Filas e Pilhas)

