Universidade do Trabalho Digital

PROGRAMAÇÃO ESTRUTURADA

Estruturas de decisão

CENTEC
INSTITUTO CENTRO DE ENSINO TECNOLÓGICO

GOVERNO DO ESTADO DO CEARÁ
Secretaria da Ciência, Tecnologia e Educação Superior

Estruturas de decisão

- Todos os dias temos que tomar várias decisões sobre o que faremos no cotidiano. Os programas não são diferentes. Eles frequentemente tomam decisões para executar suas tarefas.

- Estruturas de decisão são um recurso valioso na programação. Elas são usadas para controlar o fluxo do código na tomada de decisões de acordo com o resultado de um teste lógico como: se estiver chovendo, saia de guarda chuvas.

Universidade do Trabalho Digital

Estruturas de decisão IF

```
int dividendo = 16;

if(dividendo%2==0) {
    System.out.printf("%d é divisível por 2!",dividendo);
}
```

- A estrutura de decisão IF é bastante simples. Ela analisa um teste lógico e caso ele seja verdadeiro executa o bloco, senão o programa apenas continua.

Estrutura de decisão IF-ELSE.

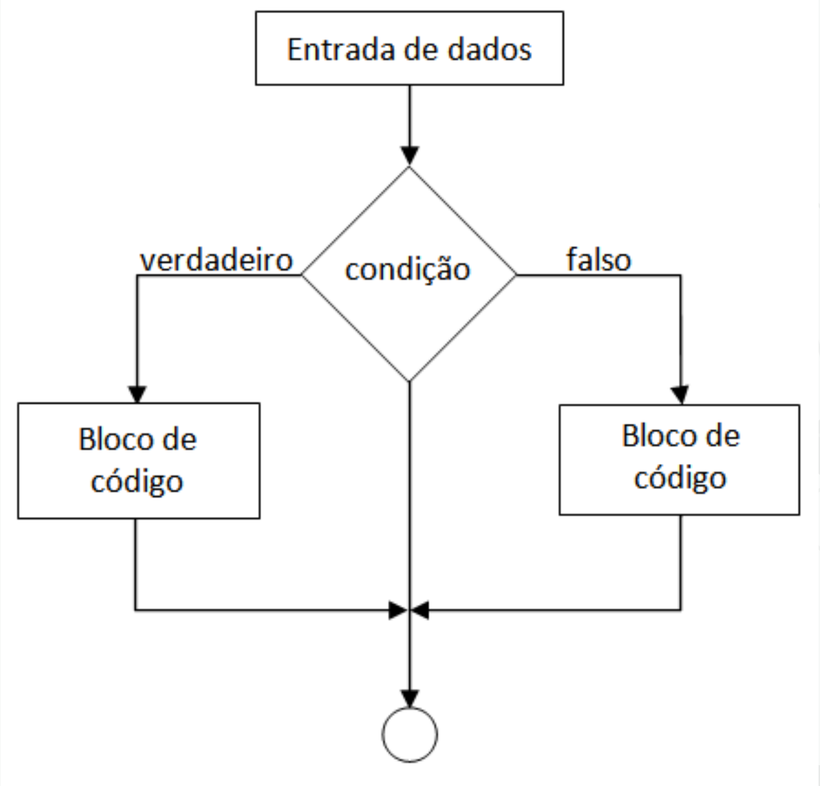


diagrama de fluxo da estrutura IF-ELSE.

Estrutura de decisão IF-ELSE

```
int dividendo = 19;

if(dividendo%2==0) {
    System.out.printf("%d é par!",dividendo);
}else {
    System.out.printf("%d não é par!",dividendo);
}
```

A estrutura de decisão IF-ELSE é mais completa, caso o testo lógico seja falso agora temos um outro bloco de código que será executado.

Estrutura de decisão switch case

- A estrutura switch case é muito utilizada para estrutura como menus, pois temos uma variável de controle que fará com que a estrutura execute um bloco de comandos, dependendo do seu valor. Caso nenhum valor corresponda ao da variável é executado o default.

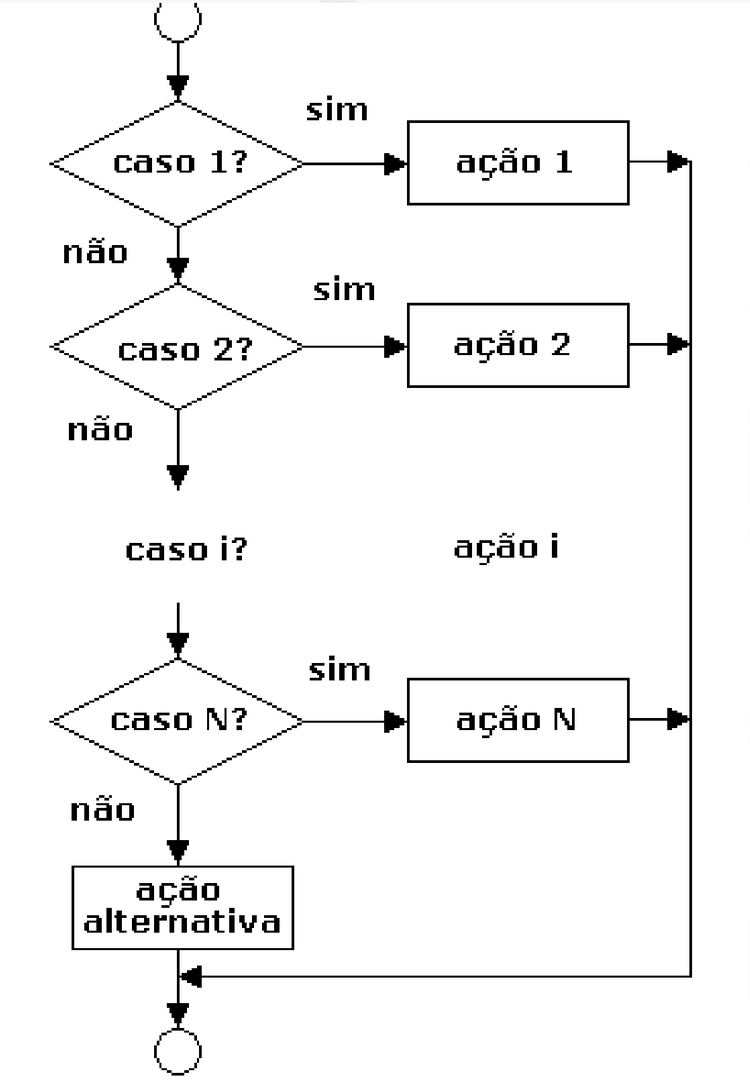


imagem fonte: IFBA

Estruturas de decisão Switch case

```
int escolha = 1;

switch(escolha) {

    case 1: System.out.println("Novo Jogo"); break;

    case 2: System.out.println("Continuar"); break;

    case 3: System.out.println("Apagar gravação"); break;

    case 4: System.out.println("Configurações"); break;

    default: System.out.println("Opção inválida");

}
```

Estruturas de repetição WHILE

```
int contador = 0;

while(contador<10) {

    System.out.println(contador);

    contador++;
}
```

As estruturas de repetição como o próprio nome sugere criam laços para que um ou mais comandos sejam repetidos sucessivamente.

A condição de parada se dá através de contadores, ou de flags booleanas, ou até mesmo através de comandos do usuário

Estruturas de repetição DO WHILE

```
int contador = 0;

while(contador<10) {

    System.out.println(contador);

    contador++;
}
```

É o inverso do while. Primeiro ele vai executar o código e só após irá verificar se a condição é verdadeira.

Essa particularidade faz com que ele execute o seu bloco de comando pelo menos uma vez, mesmo que a condição lógica seja falsa.

Estruturas de repetição FOR

```
for(int x=0; x<=10;x+=2) {

    System.out.println(x);

}
```

repetição	x
1ª	0
2ª	2
3ª	4
4ª	6
5ª	8
6ª	10
fora do laço	12

Estruturas de repetição FOR melhorado

```
char vogais[] = {'a','e','i','o','u'};

for(char letras: vogais) {

    System.out.print(letras+" ");

}
```

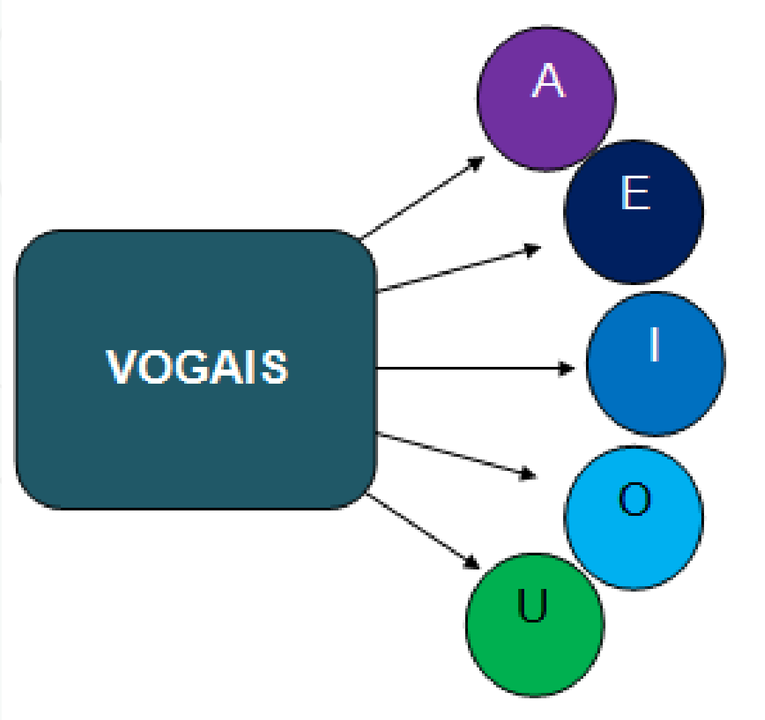
Estruturas de repetição FOR melhorado

```
char vogais[] = {'a','e','i','o','u'};

for(char letras: vogais) {

    System.out.print(letras+" ");

}
```



Funções Java

- Com retorno.
- Sem retorno void.
- Sem parâmetros.
- Com parâmetros

Ex: (String nome, int idade)

```java
public class Funcao {

    public static void main(String[] args) {

        int qtd = 8;
        double valor = 2.9;

        System.out.println(total(qtd,valor));

    }

    public static double total(int quantidade,double preco) {
        return quantidade*preco;
    }

}
```