



DEFINIÇÃO

O emprego de estruturas de decisão na Linguagem C.

PROPÓSITO

Compreender os conceitos de estrutura de decisão suportados pela Linguagem C, de forma que as aplicações desenvolvidas sejam robustas e eficientes.

OBJETIVOS

MÓDULO 1



Imagem: Shutterstock.com.

Aplicar os conceitos de estruturas de decisão simples e composta

MÓDULO 2



Imagem: Shutterstock.com.

Aplicar os conceitos de estruturas de decisão encadeada e aninhada e de múltiplas alternativas

MÓDULO 1

ESTRUTURA DE DECISÃO

No desenvolvimento de aplicações, estruturamos nossos algoritmos para realizar determinadas atividades. Uma destas estruturas é chamada **estrutura de decisão**. Esta estrutura permite que a aplicação possa ter mais de uma sequência a ser seguida, que será decidida a partir da inferência de um determinado valor. Este valor, na Linguagem C, será do tipo **booleano**, portanto, os valores poderão ser falsos (quando forem zero ou *null*) ou verdadeiros (nos demais casos).

BOOLEANO

Tipo de dado primitivo que possui dois valores, que podem ser considerados como 0 ou 1, falso ou verdadeiro. O nome é uma homenagem a George Boole (1815-1864), que definiu um sistema de lógica algébrica pela primeira vez na metade do século XIX.

Esta estrutura é conhecida como SE-ENTÃO e SE-ENTÃO-SENÃO. No primeiro caso, um determinado segmento de código será executado somente se a expressão lógica, que será inferida, for verdadeira, enquanto no segundo caso a inferência da expressão lógica determinará qual segmento de código será executado.



Imagem: belkacem aireche/Pixabay.com

Graficamente, esta estrutura é representada por um losango, em que cada quina representa:

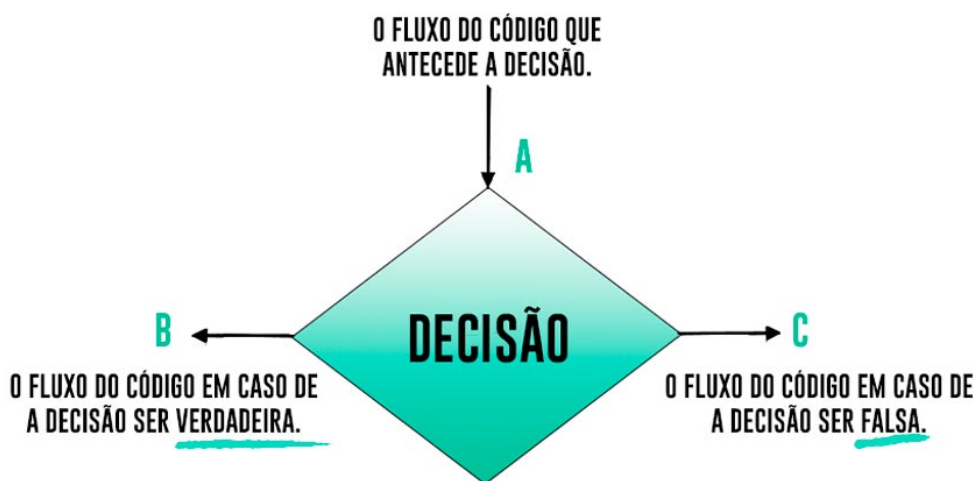
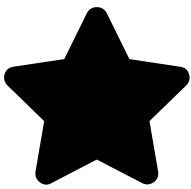


Imagem: Anderson Fernandes Pereira dos Santos

A DECISÃO CORRESPONDE A UMA EXPRESSÃO OU VARIÁVEL CUJO VALOR SERÁ ANALISADO, CONFORME JÁ CITADO.



★ EXEMPLO

Exemplo de expressões que são verdadeiras. Considere em todos os exemplos que a variável **a** é inteira e tem valor 1:

SE (a)

SE (a ==1)

SE (a > 0)

Na Linguagem C, o SE é representado pela palavra reservada `if`, assim os itens acima seriam representados nesta linguagem como:

`if(a)`

`if(a==1)`

`if(a > 0)`



📢 ATENÇÃO

É importante lembrar que a Linguagem C é sensível ao contexto, assim deve-se usar *if* com todos os caracteres minúsculos. Comandos como *If*, *iF* e *IF* não são válidos. Além disso, os parênteses utilizados acima são obrigatórios.

Mais adiante, falaremos sobre quando tivermos uma estrutura destas dentro de outra estrutura, assim chamada de aninhada, e quando duas estruturas estão posicionadas de forma sequencial, denominada de encadeada.

ESTRUTURA DE DECISÃO SIMPLES

Estrutura de decisão simples é caracterizada pela verificação de uma expressão lógica e, caso ela atenda aos requisitos estipulados, que neste caso será a expressão ser verdadeira, um determinado segmento de código é executado. Caso seja falso, ou seja, valor igual a zero, nulo ou vazio, nenhum segmento de código é executado e o programa continua no comando que sucede ao *if*.

Graficamente, esta estrutura pode ser apresentada através da figura:

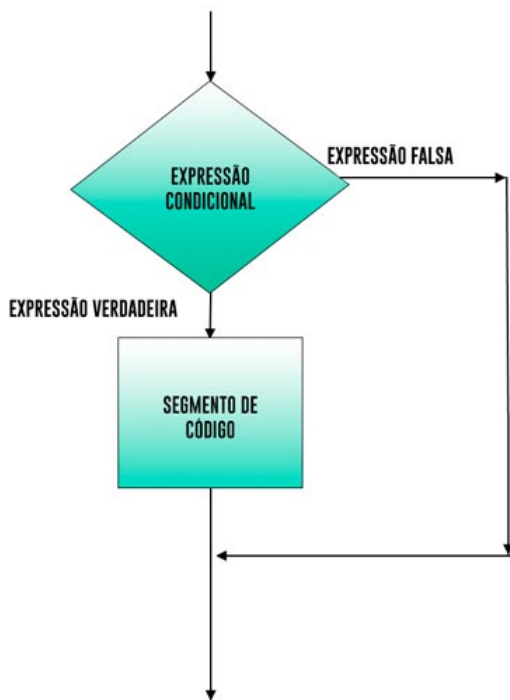


Foto: Shutterstock.com.

No losango, que representa a estrutura de decisão, temos a expressão lógica. Esta expressão será avaliada. Caso o seu valor seja verdadeiro, um bloco de comandos (segmento de código) é executado logo após a expressão ter sido calculada. Caso contrário, será dada continuidade à execução da aplicação.

Na tabela a seguir, do lado esquerdo é exibido este código representado em PORTUGOL, e do lado direito o mesmo código, usando a Linguagem C.

PORTUGOL	LINGUAGEM C
SE EXPRESSÃO_CONDICIONAL	if(expressao_condicional)
ENTÃO	{
BLOCO DE COMANDOS	bloco_de_comandos;
FIM_ENTÃO	}
FIM_SE	

1

Na primeira linha é apresentado o início do comando. No Portugol, a expressão SE representa o comando *if*.

A expressão condicional é apresentada obrigatoriamente entre parênteses. É recomendável que não haja espaço entre a palavra reservada *if* e o parêntese. Todavia é permitido, e até aconselhável, que haja espaço entre a expressão condicional e os parênteses.

PORTUGOL	LINGUAGEM C
SE EXPRESSÃO_CONDICIONAL	if(expressao_condicional)
ENTÃO	{
BLOCO DE COMANDOS	bloco_de_comandos;
FIM_ENTÃO	}
FIM_SE	

2

Na linha seguinte é apresentada a palavra ENTÃO, que interpretamos como o início do bloco de comandos, que na Linguagem C é representado pelo sinal de abre chaves {.

PORTUGOL	LINGUAGEM C
SE EXPRESSÃO_CONDICIONAL	if(expressao_condicional)
ENTÃO	{
BLOCO DE COMANDOS	bloco_de_comandos;
FIM_ENTÃO	}
FIM_SE	

3

O bloco de comandos é apresentado na sequência, porém, na Linguagem C, caso haja apenas um comando neste bloco, o uso das chaves torna-se opcional.

No caso de existirem duas ou mais instruções, estas chaves se tornam obrigatórias, uma vez que poderia haver confusão com o caso anterior.



★ EXEMPLO

Normalmente, ao preencheremos formulários na web, somos questionados se desejamos receber mais informação a respeito daquele assunto. Como isto poderia ser representado na parte da implementação do código?

```
int Flag_Deseja_Receber_Mais_Informacoes;  
if (Flag_Deseja_Receber_Mais_Informacoes){  
    Enviar_Mais_Informacoes();  
}
```

Nossa expressão condicional é uma variável do tipo inteira: `Flag_Deseja_Receber_Mais_Informacoes`. Assim, ela recebe valores inteiros.

Já vimos que na Linguagem C é considerado falso se a variável possui valores 0, *null* ou vazio; e é considerado verdadeiro no caso contrário, ou seja, se é diferente dos valores 0, *null* e vazio, dado o tipo de dado.

Neste caso, a variável é inteira, então, caso o valor dela seja 0, será considerado falso, caso possua qualquer outro valor (positivo ou negativo), será considerado verdadeiro. Outra expressão condicional equivalente poderia ser colocada neste *if*, porém, o mais usual é como foi apresentado. Um exemplo equivalente seria `Flag_Deseja_Receber_Mais_Informacoes!=0`.

Antes de resolver os exercícios propostos, assista ao vídeo abaixo e entenda mais sobre o emprego da estrutura de decisão:



CONTEXTUALIZAÇÃO DO TEMA

Neste vídeo, iremos fazer uma breve contextualização do tema.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



EXERCÍCIOS

1) MARQUE A ALTERNATIVA QUE APRESENTA A FORMA ADEQUADA DE IMPLEMENTAÇÃO DE UM CÓDIGO QUE VERIFIQUE SE UMA DIVISÃO PODE SER REALIZADA, ATRAVÉS DA AVERIGUAÇÃO DE O DIVISOR SER NULO OU NÃO. CONSIDERE AS VARIÁVEIS FLOAT A, B, C; DE ACORDO COM A EQUAÇÃO $C=A/B$.

A) `if(b==0){`
 `printf("WARNING: Divisão por zero");`
}

B) `if(a==0){ {`
 `printf("WARNING: Divisão por zero");`
}

C) `if(a==b){`
 `printf("WARNING: Divisão por zero");`
}

D) `if(a+b==0){`
 `printf("WARNING: Divisão por zero");`
}

2) EM UMA UNIVERSIDADE, UM ALUNO É CONSIDERADO APROVADO SE A MÉDIA DAS DUAS PROVAS AV1 E AV2 FOR MAIOR OU IGUAL A 7. IMPLEMENTE O SEGMENTO DO CÓDIGO, NA LINGUAGEM C, QUE INFORMARÁ AO USUÁRIO (ALUNO) SE ELE ESTÁ APROVADO. MARQUE A ALTERNATIVA QUE APRESENTA O SEGMENTO DE CÓDIGO CORRETO.

CONSIDERE:

// PRIMEIRO DEFINIMOS NOSSAS VARIÁVEIS.

FLOAT AV1, AV2, MEDIA;

//DEPOIS PROCESSAMOS A ENTRADA

//PRIMEIRO COM A INFORMAÇÃO PARA O USUÁRIO

PRINTF("INFORME A NOTA DA AV1");

//DEPOIS COM O COMANDO DE LEITURA.

SCANF("%F", &AV1);

//REPETIMOS PARA A SEGUNDA NOTA.

PRINTF("INFORME A NOTA DA AV2");

SCANF("%F", &AV2);

//CALCULAMOS AGORA A MÉDIA ARITMÉTICA.

MEDIA = (AV1 + AV2) / 2.0;

A) if(media>=7){
printf("PARABÉNS! Você está aprovado!\n");
}

B) if(media>7){
printf("PARABÉNS! Você está aprovado!\n");
}

C) if(media==7){
printf("PARABÉNS! Você está aprovado!\n");
}

```
D) if(media<=7){  
printf("PARABÉNS! Você está aprovado!\n");  
}
```

GABARITO

1) Marque a alternativa que apresenta a forma adequada de implementação de um código que verifique se uma divisão pode ser realizada, através da averiguação de o divisor ser nulo ou não. Considere as variáveis float a, b, c; de acordo com a equação $c=a/b$.

A alternativa **"A "** está correta.

A opção correta é a letra A, uma vez que, para a divisão ser possível, é necessário que o divisor seja não nulo.

2) Em uma universidade, um aluno é considerado aprovado se a média das duas provas AV1 e AV2 for maior ou igual a 7. Implemente o segmento do código, na Linguagem C, que informará ao usuário (aluno) se ele está aprovado. Marque a alternativa que apresenta o segmento de código correto.

Considere:

```
// Primeiro definimos nossas variáveis.
```

```
float av1, av2, media;
```

```
//Depois processamos a entrada
```

```
//Primeiro com a informação para o usuário
```

```
printf("Informe a nota da AV1");
```

```
//Depois com o comando de leitura.
```

```
scanf("%f", &av1);
```

```
//Repetimos para a segunda nota.
```

```
printf("Informe a nota da AV2");
```

```
scanf("%f", &av2);
```

```
//Calculamos agora a média aritmética.
```

```
media = (av1 + av2) / 2.0;
```

A alternativa **"A "** está correta.

O enunciado cita que a média deve ser maior ou igual a 7, portanto, a única alternativa que utiliza o condicional correto é a letra A, \geq .

3) CONSIDERE O SEGMENTO DE CÓDIGO ABAIXO, QUE FAZ PARTE DE UMA APLICAÇÃO QUE VERIFICA SE O ESTOQUE DE UM DETERMINADO PRODUTO EM UM SUPERMERCADO ALCANÇOU A QUANTIDADE MÍNIMA OU NÃO.

CASO TENHA ALCANÇADO A QUANTIDADE MÍNIMA, É EMITIDA UMA MENSAGEM PARA O FORNECEDOR ENTREGAR MAIS ITENS DAQUELE PRODUTO. CASO O ESTOQUE MÍNIMO DO PRODUTO SEJA ZERO, ENTÃO NÃO É REALIZADO NOVO PEDIDO. CASO O ESTOQUE ATUAL DO PRODUTO SEJA ZERO, UMA MENSAGEM É EMITIDA PARA O FORNECEDOR ENTREGAR ALGUNS ITENS DAQUELE PRODUTO.

PARA CADA PRODUTO, O SEU ESTOQUE MÍNIMO É CALCULADO EM OUTRA PARTE DO CÓDIGO E ARMAZENADO NA VARIÁVEL ESTOQUE_MINIMO_PRODUTO. DA MESMA FORMA, O ESTOQUE ATUAL É CALCULADO EM OUTRA PARTE DO CÓDIGO E ARMAZENADO NA VARIÁVEL ESTOQUE_ATUAL_PRODUTO. ASSIM, ESTAS VARIÁVEIS, AO COMPOREM A EXPRESSÃO LÓGICA, JÁ POSSUIRÃO OS VALORES PARA UM DETERMINADO PRODUTO.

O SEGMENTO DE CÓDIGO É APRESENTADO ABAIXO.

```
IF(EXPRESSAO_CONTROLE_ESTOQUE){  
    EMITIR_MENSAGEM_NOVO_PEDIDO();  
    PRINTF("ENVIADA MENSAGEM PARA FORNECEDOR");  
}
```

CONSIDERANDO AS PREMISSAS INFORMADAS ANTERIORMENTE PARA AS VARIÁVEIS ESTOQUE_ATUAL_PRODUTO E ESTOQUE_MINIMO_PRODUTO, MARQUE A OPÇÃO QUE APRESENTA O VALOR ADEQUADO PARA A VARIÁVEL EXPRESSÃO_CONTROLE_ESTOQUE, BASEADO NAS DUAS VARIÁVEIS ANTERIORES.

A) expressão_controle_estoque = (estoque_minimo_produto>0) || (estoque_minimo_produto > estoque_atual_produto) || (estoque_atual_produto==0)

B) expressão_controle_estoque = (estoque_minimo_produto>0) && ((estoque_minimo_produto > estoque_atual_produto) && (estoque_atual_produto==0))

C) expressão_controle_estoque = (estoque_minimo_produto>0) && ((estoque_minimo_produto > estoque_atual_produto) || (estoque_atual_produto==0))

D) expressão_controle_estoque = (estoque_minimo_produto>0) || ((estoque_minimo_produto > estoque_atual_produto) && (estoque_atual_produto==0))

4) MARQUE A OPÇÃO QUE APRESENTA UM EXEMPLO DE OPERAÇÃO QUE NÃO PODERÁ SER UTILIZADA NO SEGMENTO DE CÓDIGO ABAIXO.

INT A = 1, B = 2, C = 3;

```
IF(EXPRESSÃO){  
    PRINTF(“OPERAÇÃO REALIZADA COM SUCESSO”);  
}
```

A) if(“PI”){

B) if(a==2){

C) if(a-2){

D) if(!a=3){

GABARITO

3) Considere o segmento de código abaixo, que faz parte de uma aplicação que verifica se o estoque de um determinado produto em um supermercado alcançou a quantidade mínima ou não.

Caso tenha alcançado a quantidade mínima, é emitida uma mensagem para o fornecedor entregar mais itens daquele produto. Caso o estoque mínimo do produto seja zero, então não é realizado novo pedido. Caso o estoque atual do produto seja zero, uma mensagem é emitida para o fornecedor entregar alguns itens daquele produto.

Para cada produto, o seu estoque mínimo é calculado em outra parte do código e armazenado na variável `estoque_minimo_produto`. Da mesma forma, o estoque atual é calculado em outra parte do código e armazenado na variável `estoque_atual_produto`. Assim, estas variáveis, ao

comporem a expressão lógica, já possuirão os valores para um determinado produto.

O segmento de código é apresentado abaixo.

```
if(expressao_controle_estoque){  
    Emitir_Mensagem_Novo_Pedido();  
    printf("Enviada mensagem para fornecedor");  
}
```

Considerando as premissas informadas anteriormente para as variáveis `estoque_atual_produto` e `estoque_minimo_produto`, marque a opção que apresenta o valor adequado para a variável `expressão_controle_estoque`, baseado nas duas variáveis anteriores.

A alternativa "C " está correta.

Os requisitos são:

I. Caso tenha alcançado a quantidade mínima, é emitida uma mensagem para o fornecedor entregar mais itens daquele produto.

`estoque_minimo_produto > estoque_atual_produto`

II. Caso o estoque mínimo do produto seja zero, então não é realizado novo pedido.

`estoque_minimo_produto > 0`

III. Caso o estoque atual do produto seja zero, uma mensagem é emitida para o fornecedor entregar alguns itens daquele produto.

`estoque_minimo_produto==0`

A resposta certa, portanto, é a **alternativa C**.

4) Marque a opção que apresenta um exemplo de operação que não poderá ser utilizada no segmento de código abaixo.

```
int a = 1, b = 2, c = 3;
```

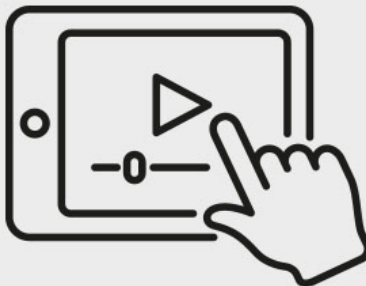
```
if(expressão){  
    printf("Operação realizada com sucesso");  
}
```

A alternativa "A " está correta.

As expressões b, c e d estão corretas, mas a expressão a não está. Um conjunto de caracteres não pode ser um condicional para o operador *if*.

TEORIA NA PRÁTICA

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Estruturas de decisão correspondem a uma prática comum e estão presentes em praticamente todas as aplicações. Estas estruturas podem ser do tipo simples, composta, encadeada ou aninhada. Esta representação é tão comum, que até mesmo filmes que possuem algum teor computacional retratam esta situação.

Uma das formas clássicas apresentadas está no filme *Matrix*, no qual o personagem Morpheus apresenta duas opções para o personagem Neo.

ESTRUTURA DE DECISÃO COMPOSTA

Como vimos anteriormente, a estrutura de decisão simples permite que um bloco seja executado caso uma expressão lógica seja verdadeira.

Agora acrescentaremos um novo bloco, que será executado no caso de a condição ser falsa. Assim, estamos nos referindo à estrutura SE-ENTÃO-SENÃO, conforme exibido na figura a seguir.

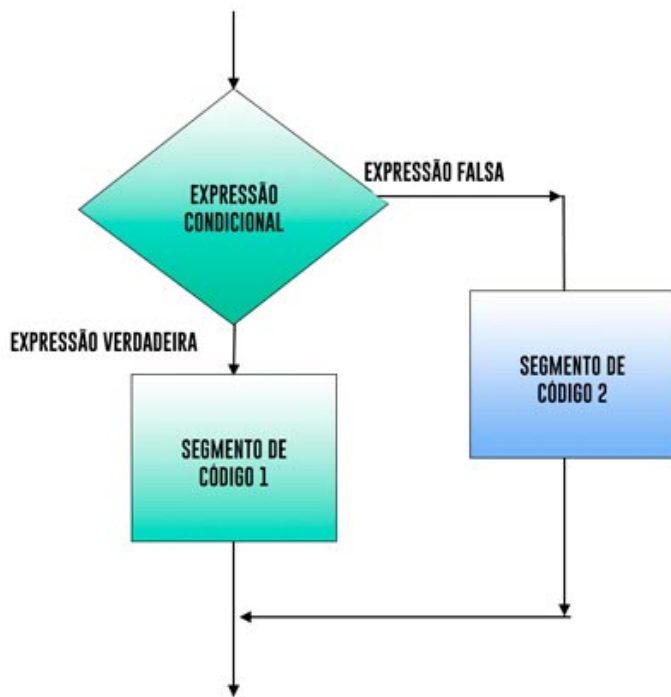


Imagem: Anderson Fernandes Pereira dos Santos.

A PRINCIPAL DIFERENÇA DESTE TIPO DE ESTRUTURA CONDICIONAL E A ANTERIOR RESIDE NO BLOCO MARCADO EM AZUL NA IMAGEM ANTERIOR. NESTE CASO, QUANDO A EXPRESSÃO CONDICIONAL FOR AVALIADA COMO FALSA, O BLOCO AZUL SERÁ EXECUTADO.

Dessa forma, quando uma expressão condicional é avaliada, caso o seu valor seja verdadeiro, um bloco de comandos 1 é executado, caso contrário, será executado o bloco de comandos 2.

Após a execução do bloco de comandos selecionado pela expressão, é dada continuidade à execução da aplicação.

Na tabela a seguir, do lado esquerdo é exibido este código representado em PORTUGOL, e do lado direito o mesmo código, usando a Linguagem C. A principal diferença perante o caso anterior está relacionada nas linhas marcadas em amarelo, que representam o quadrado azul do diagrama. Assim, caso a expressão condicional seja falsa, o código que será executado será o presente na estrutura *else*.



📢 ATENÇÃO

A mesma regra com relação à quantidade de instruções e as chaves é utilizada neste caso. Portanto, se o bloco possuir apenas um comando, as chaves são opcionais, porém normalmente utilizadas. No caso de haver mais de uma instrução, estas chaves tornam-se obrigatórias.

PORTUGOL	LINGUAGEM C
SE EXPRESSÃO_CONDICIONAL	if(expressao_condicional)
ENTÃO	{
BLOCO DE COMANDOS 1	bloco_de_comandos_1;
FIM_ENTÃO	}
SENÃO	else
	{
BLOCO DE COMANDOS 2	bloco_de_comandos_2;
FIM_SENÃO	}
FIM_SE	

Imagem: Anderson Fernandes Pereira dos Santos.

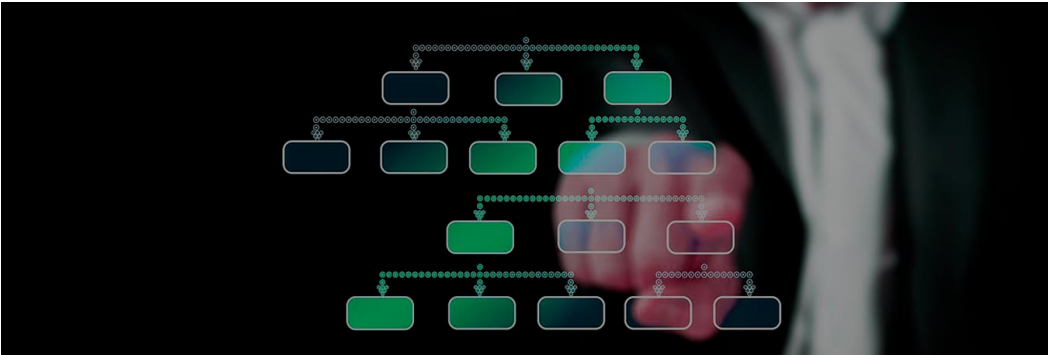


Imagem: Shutterstock.com

📷 O DBT trabalha a aceitação através de metas comportamentais

TEORIA NA PRÁTICA

No final da década de 1990, a internet foi desenvolvida nos laboratórios do CERN por **Tim Berners-Lee**. Desde aquele momento em que passamos a estar globalmente conectados com todas as pessoas do mundo, o avanço da tecnologia caminha a passos largos. Nos últimos anos, a Inteligência Artificial passou a fazer parte do nosso dia a dia. Quer seja em nosso aplicativo de streaming de música ou vídeo, quer seja no percurso que nos é sugerido para ir ao trabalho, nos tornamos usuários ferrenhos desta tecnologia.

TIMOTHY JOHN BERNERS-LEE

É um físico britânico, cientista da computação e professor do MIT. É o criador da World Wide Web.

MAS, O QUE, DE FATO, É INTELIGÊNCIA ARTIFICIAL?

De acordo com Adhikari (2019), Inteligência Artificial é uma aplicação que pode ser implementada com uma grande coleção de instruções *if-e/se* ou com um modelo estatístico complexo e parâmetros avançados, ambos com a mesma finalidade: simulação da inteligência humana em máquinas.

VEM QUE EU TE EXPLICO!

Os vídeos a seguir abordam os assuntos mais relevantes do conteúdo que você acabou de estudar.

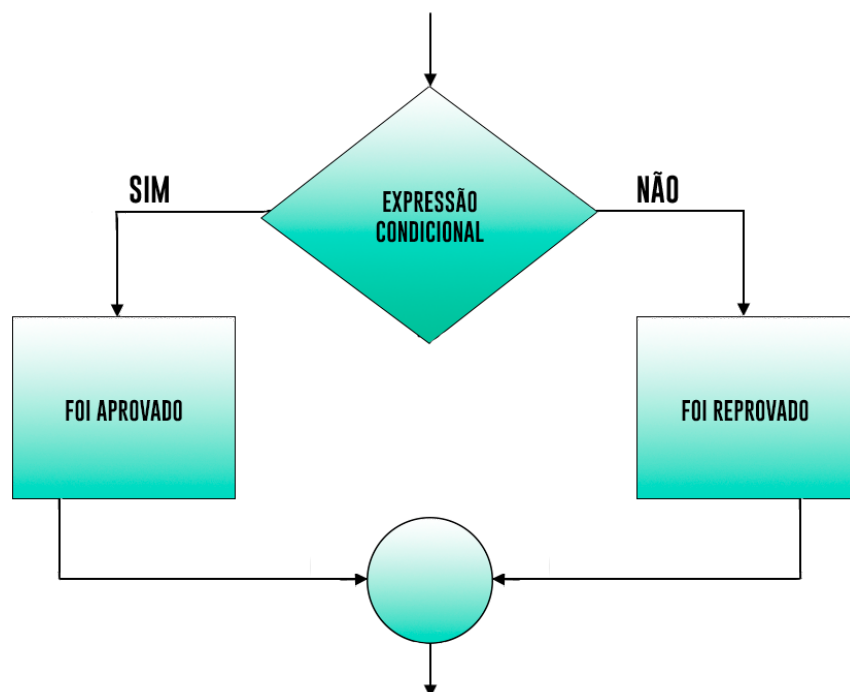
Exercícios discussão do exercício 2 nas linguagens Java e Python

Exercícios discussão do exercício 4 nas linguagens Java e Python

Estrutura de decisão composta

VERIFICANDO O APRENDIZADO

1) (BASEADA EM CESPE - 2017 - TRT - 7ª REGIÃO (CE) - ANALISTA JUDICIÁRIO - TECNOLOGIA DA INFORMAÇÃO)



A ESTRUTURA LÓGICA PRESENTE NO DIAGRAMA APRESENTADO É DO TIPO:

- A) SE ENTÃO
- B) CASO SELECIONE
- C) CASO REPITA
- D) SE ENTÃO SENÃO

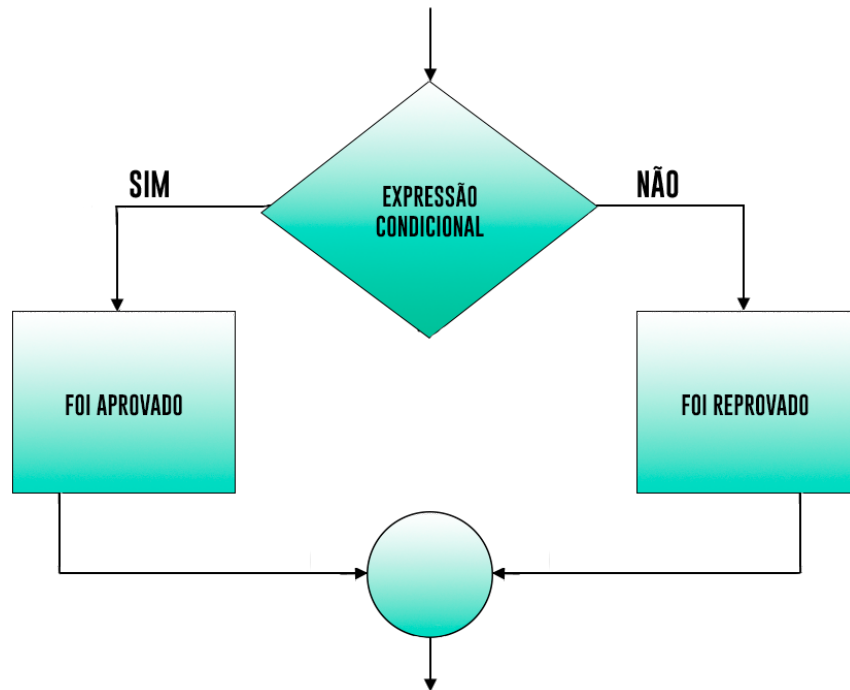
2) (BASEADA EM FCC - 2010 - SP - AGENTE DE DEFENSORIA - ANALISTA DE SISTEMAS) MARQUE A ALTERNATIVA QUE APRESENTA O COMANDO UTILIZADO NA ESTRUTURA DE DECISÃO COMPOSTA, OU SEJA, QUE POSSUA A OPÇÃO A SER EXECUTADA CASO A CONDIÇÃO SEJA VERDADEIRA E CASO A CONDIÇÃO SEJA FALSA.

- A) for(;;)
- B) if{}else{}
- C) while(){}

D) do{}while()

GABARITO

1) (Baseada em CESPE - 2017 - TRT - 7ª Região (CE) - Analista Judiciário - Tecnologia da Informação)



A estrutura lógica presente no diagrama apresentado é do tipo:

A alternativa "D " está correta.

A estrutura apresentada é do tipo se então (indicado com o sim na figura) senão (indicado com o não da figura).

2) (Baseada em FCC - 2010 - SP - Agente de Defensoria - Analista de Sistemas) Marque a alternativa que apresenta o comando utilizado na estrutura de decisão composta, ou seja, que possua a opção a ser executada caso a condição seja verdadeira e caso a condição seja falsa.

A alternativa "B " está correta.

Opção correta, letra B. A única opção que possui estrutura de condição é a alternativa b. As outras opções mostram estruturas de repetição que possuem características e aplicações diferentes em programação.

MÓDULO 2

- ⦿ **Aplicar os conceitos de estruturas de decisão encadeada e aninhada e de múltiplas alternativas**

INTRODUÇÃO

Cada bloco que compõe um *if*, seja simples ou composto, é formado por um ou mais comandos. Conforme já mencionado, caso seja apenas um comando, o uso de chaves é opcional. Caso seja utilizado mais de um comando, o uso de chaves torna-se obrigatório, para que seja determinado o bloco de comandos que ficará englobado pelo *if* ou *else*.

Mas, ainda poderão ocorrer outros casos:

Quando um dos comandos que compõe o *if* (ou *else*) for outro comando *if* (simples ou composto), temos o que é chamado de *if* aninhado

Quando dois comandos *ifs*
forem sequenciais,
teremos o caso de *if* encadeado

Descreveremos estes dois tipos nas próximas seções.

ESTRUTURAS DE DECISÃO ANINHADAS

Este é o primeiro caso descrito acima, quando um dos comandos que compõem um *if* (ou *else*) é outro comando *if*, podendo estes comandos serem simples ou compostos. Ou seja, as estruturas são dispostas umas dentro das outras. A seguir, apresentamos uma estrutura de decisão composta internamente de uma estrutura de decisão simples.

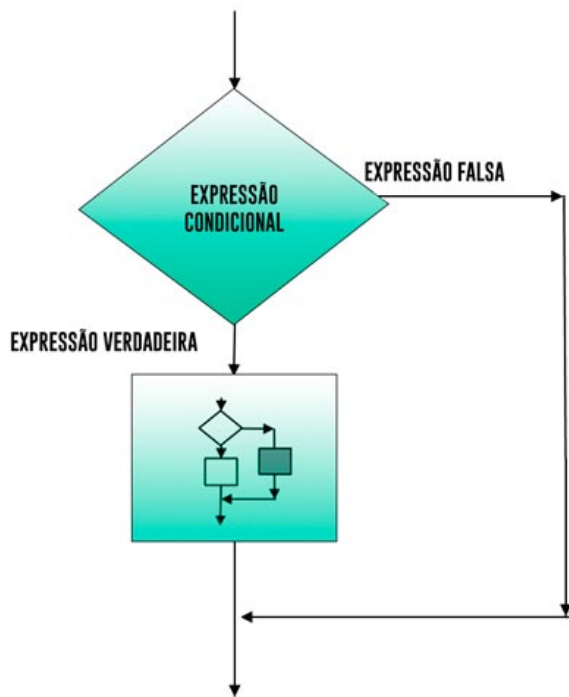


Imagem: Anderson Fernandes Pereira dos Santos.

Agora que já vimos a representação gráfica, partiremos para a representação na Linguagem C. Conforme já mencionado, temos um *if* composto internamente a um *if* simples, assim, esta figura representa o seguinte código em C:

```

if(EXPRESSAO_CONDICIONAL_1){
  if(EXPRESSAO_CONCIDIONAL_2){
    BLOCO_INSTRUCAO_1;
  } else {
    BLOCO_INSTRUCAO_2;
  }
}

```

Se o *if* interno fosse simples, o código já seria expresso por:

```

if(EXPRESSAO_CONDICIONAL_1){
  if(EXPRESSAO_CONCIDIONAL_2){
    BLOCO_INSTRUCAO_1;
  }
}

```

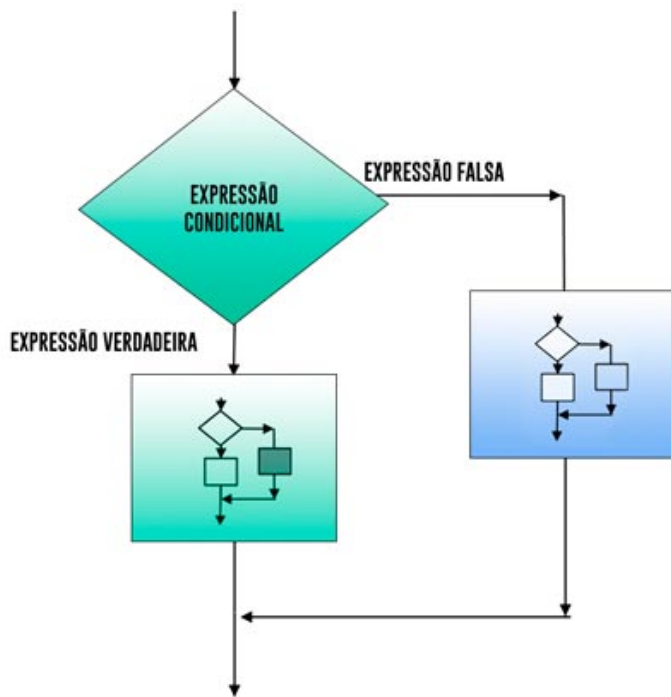


Imagem: Anderson Fernandes Pereira dos Santos.

No caso de as duas estruturas de decisão serem compostas, estas são apresentadas dentro das instruções da estrutura de decisão. Convém salientar que não há limitação com relação à quantidade de estruturas de decisão que são dispostas internamente, independentemente do seu tipo, quer sejam simples ou compostas.

Um exemplo de estrutura de decisão aninhada, já na Linguagem C, semelhante ao diagrama anterior, é exibido a seguir.

```

if(EXPRESSAO_CONDICIONAL_1){
  if(EXPRESSAO_CONCIDIONAL_2){
    BLOCO_INSTRUCAO_1;
  } else {
    BLOCO_INSTRUCAO_2;
  }
} else {
  EXPRESSAO_CONCIDIONAL_3
  BLOCO_INSTRUCAO_3;
} else {
  BLOCO_INSTRUCAO_4;
}
}

```

De forma semelhante, no caso de um *if* simples ser um comando de uma estrutura *e/se* na Linguagem C, seria representado como:

```
if(EXPRESSAO_CONDICIONAL_1){  
    BLOCO_INSTRUCAO_1;  
}  
else {  
    if(EXPRESSAO_CONCIDIONAL_2){  
        BLOCO_INSTRUCAO_2;  
    }  
}
```

Ou ainda, no caso do *if* simples ser um comando do *if* na Linguagem C, seria representado como:

```
if(EXPRESSAO_CONDICIONAL_1){  
    if(EXPRESSAO_CONCIDIONAL_2){  
        BLOCO_INSTRUCAO_1;  
    }  
} else {  
    BLOCO_INSTRUCAO_2;  
}
```

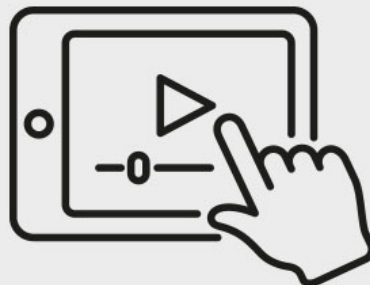
Antes de resolver os exercícios propostos, assista ao vídeo abaixo e entenda mais sobre o emprego da estrutura de decisão aninhada:



CONTEXTUALIZAÇÃO DO TEMA

Neste vídeo, iremos fazer uma breve contextualização do tema.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



EXERCÍCIOS

1) (BASEADO EM IFAL – 2014) PARA QUE FAIXA DE VALORES DA VARIÁVEL X O SEGUINTE TRECHO DE CÓDIGO IMPRIME A LETRA C?

```
IF( X <= 200)  
IF (X < 100)  
IF (X < 0) PRINTF("A")  
ELSE PRINTF("B")  
ELSE PRINTF("C")  
ELSE PRINTF("D")
```

- A) $0 < x < 100$**
- B) $x \leq 100$**
- C) $100 \leq x \leq 200$**
- D) $x > 200$**

2) NOSSOS CARROS PODEM SER ABASTECIDOS COM ÁLCOOL OU COM GASOLINA, MAS O PREÇO E O RENDIMENTO SÃO BEM DIFERENTES. ENTÃO, COMO DEVEMOS ABASTECER? DIZ A SABEDORIA POPULAR QUE, PARA QUE SEJA RECOMENDÁVEL O ABASTECIMENTO COM ÁLCOOL, O PREÇO DESTES COMBUSTÍVEL DEVERÁ SER ATÉ 70% DO PREÇO DA GASOLINA. ASSIM, MARQUE A OPÇÃO QUE APRESENTA O COMPLEMENTO DO CÓDIGO DE UM PROGRAMA EM C QUE, APÓS TER SIDO FORNECIDO PELO USUÁRIO O PREÇO DO LITRO DOS DOIS COMBUSTÍVEIS, INFORME COM QUAL DEVEREMOS ABASTECER.

CÓDIGO EXISTENTE:

```
// DEFINIÇÃO DAS VARIÁVEIS QUE SERÃO UTILIZADAS.  
FLOAT PREÇO_ALCOOL, PREÇO_GASOLINA, FATOR_DECISAO;  
//ENTRADADA DOS DADOS  
PRINTF("FONEÇA O PREÇO DO ÁLCOOL:");  
SCANF("%f", &PREÇO_ALCOOL);  
PRINTF("FONEÇA O PREÇO DA GASOLINA:");  
SCANF("%f", &PREÇO_GASOLINA);
```

//PROCESSAMENTO E SAÍDA.

PRINTF("ABASTEÇA COM ");

A) if(preco_alcool>0)

if((preco_alcool/preco_gasolina)<=0,7)

printf(" Álcool\n");

else

printf(" Gasolina\n");

B) if(preco_gasolina>0)

if((preco_gasolina/preco_alcool)<=0,7)

printf(" Álcool\n");

else

printf(" Gasolina\n");

C) if(preco_alcool>0)

if((preco_gasolina/preco_alcool)<=0,7)

printf(" Álcool\n");

else

printf(" Gasolina\n");

D) if(preco_gasolina>0)

if((preco_alcool/preco_gasolina)<=0,7)

printf(" Álcool\n");

else

printf(" Gasolina\n");

GABARITO

1) (Baseado em IFAL – 2014) Para que faixa de valores da variável x o seguinte trecho de código imprime a letra C?

if(x <= 200)

if (x < 100)

if (x < 0) printf("A")

else printf("B")

else printf("C")

else printf("D")

A alternativa **"C "** está correta.

A resposta correta é a letra C. Ela será impressa se o primeiro *if* for verdadeiro e o segundo *if* for falso. No primeiro, *if* indica que $x \leq 200$ e o segundo *if* indica que o valor deverá ser a negação de $x < 100$, que neste caso é $x \geq 100$. Combinando as duas expressões, temos que $x \geq 100$ e $x \leq 200$, resultando na letra C.

2) Nossos carros podem ser abastecidos com álcool ou com gasolina, mas o preço e o rendimento são bem diferentes. Então, como devemos abastecer? Diz a sabedoria popular que, para que seja recomendável o abastecimento com álcool, o preço deste combustível deverá ser até 70% do preço da gasolina. Assim, marque a opção que apresenta o complemento do código de um programa em C que, após ter sido fornecido pelo usuário o preço do litro dos dois combustíveis, informe com qual deveremos abastecer.

Código existente:

```
// definição das variáveis que serão utilizadas.  
float preco_alcool, preco_gasolina, fator_decisao;  
//entrada de dados  
printf("Foneça o preço do álcool:");  
scanf("%f", &preco_alcool);  
printf("Foneça o preço da gasolina:");  
scanf("%f", &preco_gasolina);  
  
//processamento e saída.  
printf("Abasteça com ");
```

A alternativa "D " está correta.

A resposta correta é a letra D. Como estamos falando de divisão dos dois valores, obrigatoriamente o denominador, que neste caso é o preço da gasolina, não poderá ser nulo. Assim, automaticamente a alternativa A e B estão incorretas porque estão sendo testados os numeradores. A alternativa B e C está incorreta, pois a expressão que está sendo avaliada está incorreta, o correto é dividir o preço do álcool pelo da gasolina.

ESTRUTURAS DE DECISÃO ENCADEADAS

Estruturas encadeadas são dispostas de forma sequencial. Independentemente de serem simples ou complexas, estas são dispostas sequencialmente, então, ao término de uma estrutura, é considerada a entrada da próxima estrutura, conforme o segmento de código abaixo.

```
if(EXPRESSAO_CONDICIONAL_1){  
    BLOCO_INSTRUCAO_1;  
}  
else  
if(EXPRESSAO_CONDICIONAL_2){  
    BLOCO_INSTRUCAO_3;  
}  
else {  
    BLOCO_INSTRUCAO_4;  
}
```

Neste exemplo, uma estrutura *if* composta está disposta logo após o *else* da estrutura anterior. Se esta segunda estrutura fosse um *if* simples, na Linguagem C seria representado como:

```
if(EXPRESSAO_CONDICIONAL_1){  
    BLOCO_INSTRUCAO_1;  
}  
else  
if(EXPRESSAO_CONDICIONAL_2){  
    BLOCO_INSTRUCAO_3;  
}
```

Deve-se notar que em todos os exemplos apresentados até este ponto, BLOCO_INSTRUCAO pode ser representado por um ou mais comandos. Caso seja apenas um comando, o uso das chaves é opcional. Caso possua mais de um comando, o uso das chaves torna-se obrigatório.

Outro ponto que merece destaque é que um comando *if*, conforme o código abaixo, é dito ser apenas um comando, e os blocos de instrução que estão contidos neste código, mesmo que possuam mais de um comando, compõem apenas um, e assim não precisam de chaves antes e depois do *if*.

```
if(EXPRESSAO_CONDICIONAL_1){  
    BLOCO_INSTRUCAO_1;  
}  
else  
    BLOCO_INSTRUCAO_2;  
}
```

EXERCÍCIOS

**3) (BASEADO EM FCC – TRT – 9ª REGIÃO - ANALISTA JUDICIÁRIO -
TECNOLOGIA DA INFORMAÇÃO)
CONSIDERE O SEGMENTO DE CÓDIGO ABAIXO:**

```
#INCLUDE <STDIO.H>

INT MAIN(INT ARGC, CHAR *ARGV[]){
CHAR SEXO;
PRINTF("QUAL É O SEU SEXO? (F/M) ");
SCANF("%C", &SEXO);
IF(SEXO == 'F' || SEXO == 'f')
PRINTF("VOCÊ É DO SEXO FEMININO. ");
ELSE
IF(SEXO == 'M' || SEXO == 'm')
PRINTF("VOCÊ É DO SEXO MASCULINO. ");
ELSE
PRINTF( "VOCÊ DIGITOU UM VALOR DE SEXO INVALIDO ");
}
```

SOBRE O CÓDIGO ACIMA, É CORRETO AFIRMAR QUE:

- A) A lógica do algoritmo está comprometida pela falta de um comando de repetição.
- B) Em vez de utilizar comandos de decisão aninhados, deveria ter sido usado um único comando de seleção múltipla, por isso a lógica ficou comprometida.
- C) Se for digitada uma letra maiúscula 'F' ou minúscula 'f' será impresso "Você é do sexo feminino".
- D) Somente se a letra digitada para o sexo for diferente de 'F', 'f', 'M', 'm' é que a frase "Você digitou um valor de sexo inválido" será impressa.

GABARITO

3) (Baseado em FCC – TRT – 9ª Região - Analista Judiciário - Tecnologia da Informação)

Considere o segmento de código abaixo:

```
#include <stdio.h>

int main(int argc, char *argv[]){
char sexo;

printf("Qual é o seu sexo? (f/m) ");
scanf("%c", &sexo);
if(sexo == 'f' || sexo == 'F')
printf("Você é do sexo feminino. ");
else
```

```

if(sexo == 'm' || sexo == 'M')
printf("Você é do sexo masculino. ");
else
printf( "Você digitou um valor de sexo invalido ");
}

```

Sobre o código acima, é correto afirmar que:

A alternativa "D " está correta.

A letra correta é a D. No desenvolvimento de uma aplicação, não há nenhuma obrigatoriedade de usar estrutura de repetição. A lógica não fica comprometida caso se deseje utilizar uma quantidade específica de *ifs*. O código demonstra que, se o usuário digitar 'f' ou 'F', sairá impresso "Você é do sexo feminino.", 'm' ou 'M', "Você é do sexo masculino." e "Você digitou um valor de sexo inválido" em qualquer outro caso.

OPERADOR TERNÁRIO

A Linguagem C possui um operador que funciona de forma bem semelhante a um *if*. Este operador utiliza os símbolos de (?) e de (:) para representar a expressão lógica que é utilizada e os valores que comporão o campo do *if* e o campo do *else*, fazendo uma comparação com um *if* composto.

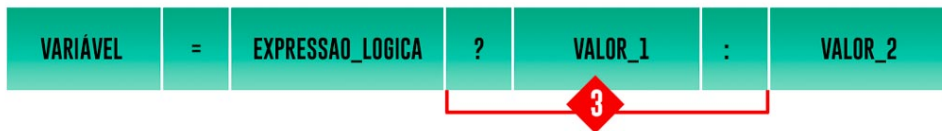
Este operador, conhecido como operador ternário por possuir três campos, é apresentado na figura abaixo.



Esta expressão deverá retornar VERDADEIRO ou FALSO. No caso da Linguagem C, o valor falso deverá ser 0, *null* ou vazio. E verdadeiro será qualquer valor diferente deste.



Valores que serão retornados por este operador. O primeiro será retornado quando o operador for verdadeiro e o segundo no caso de falso.



Os símbolos (?) e (:) deverão ser utilizados nesta ordem, pois, caso contrário, serão identificados pelo compilador como um erro sintático.



VARIAVEL corresponde à variável que receberá os valores, VALOR_1 e VALOR_2, do operador ternário. Assim, caso EXPRESSAO_LOGICA seja verdadeira, VALOR_1 será atribuído à VARIAVEL, caso contrário, VALOR_2 é que será atribuído.

Por exemplo, considere o código abaixo:

```
int a, b, c, d, e;
```

```
a=1;
```

```
b=2;
```

```
c=3;
```

```
d=4;
```

```
e=(a>b)?c:d;
```

Neste exemplo, temos que **a > b** é falso, pois **a = 1** e **b = 2**. Portanto, conforme já explicado anteriormente, a variável **e** irá receber o valor **d**.

Caso o comando fosse reescrito como `e=(b>a)?c:d;` teríamos o valor **c** sendo atribuído à variável **e**.

OPERADOR TERNÁRIO TAMBÉM PERMITE QUE SEJA REALIZADO ALINHAMENTO, PORÉM NÃO É

UMA BOA PRÁTICA DE PROGRAMAÇÃO.

ESTRUTURAS DE MÚLTIPLAS ALTERNATIVAS

Esta estrutura, também conhecida como **switch-case**, permite que seja criada uma estrutura condicional que verificará o valor de uma variável de controle, no código abaixo identificado por **VARIAVEL**, e confrontará a determinados valores A, B e C para que sejam executados os blocos de instrução **BLOCO_INSTRUCAO_1**, **BLOCO_INSTRUCAO_2**, **BLOCO_INSTRUCAO_3** e **BLOCO_INSTRUCAO_4**.

```
switch(VARIAVEL){  
case A: BLOCO_INSTRUCAO_1;  
break;  
case B: BLOCO_INSTRUCAO_2;  
break;  
case c: BLOCO_INSTRUCAO_3;  
break;  
default: BLOCO_INSTRUCAO_4;  
}
```

O campo variável precisará, obrigatoriamente, ser do tipo **char**, **int** e **long**. O que significa que os valores A, B e C precisam também ser deste tipo.

Uma vez que **VARIAVEL** seja igual a A, por exemplo, a sequência de execução de tarefas continua até que o comando **break** seja encontrado. Assim, é executado **BLOCO_INSTRUCAO_1**. Caso a estrutura acima fosse conforme o segmento abaixo, no mesmo valor de **VARIAVEL** sendo igual a A, o resultado seria a execução de **BLOCO_INSTRUCAO_1** e **BLOCO_INSTRUCAO_2**.

Isto porque o comando **break** não estava presente.

```
switch(VARIAVEL){  
case A: BLOCO_INSTRUCAO_1;  
case B: BLOCO_INSTRUCAO_2;  
break;  
case c: BLOCO_INSTRUCAO_3;  
break;  
default: BLOCO_INSTRUCAO_4;  
}
```


OUTRO PONTO IMPORTANTE É QUE NO CASO DE VARIÁVEL NÃO SER IGUAL A NENHUMA DAS OPÇÕES, ENTÃO O COMANDO DEFAULT É EXECUTADO. NO ENTANTO, DEVE-SE SALIENTAR QUE ESTE COMANDO É OPCIONAL.

EXERCÍCIOS

4) MARQUE A OPÇÃO QUE APRESENTA APENAS TIPOS VÁLIDOS QUE PODEM SER EMPREGADOS EM UMA ESTRUTURA DO TIPO SWITCH:

- A) int, char e long
- B) int, double e char
- C) float, double e char
- D) int, float e double

GABARITO

4) Marque a opção que apresenta apenas tipos válidos que podem ser empregados em uma estrutura do tipo switch:

A alternativa **"A "** está correta.

Conforme visto no item estruturas de múltiplas alternativas, os tipos que podem ser usados no switch são **int**, **char** e **long**. Assim, as letras B, C e D estão incorretas. B está incorreta por conter o tipo **double**, C e D por conter os tipos **float** e **double**.

TEORIA NA PRÁTICA

Normalmente, o comando switch é utilizado quando são ofertadas aos usuários diversas opções para que eles possam escolher, geralmente, no decorrer do preenchimento de um formulário, que hoje em dia está disponível na plataforma web ou na plataforma para dispositivos móveis. Estas opções podem ser apresentadas ao usuário em formato texto, assemelhando-se muito à estrutura da Linguagem C.

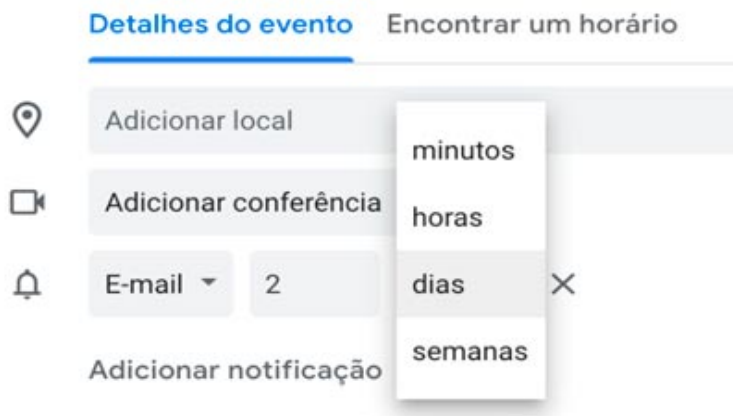


Imagem: Shutterstock.com.

Neste exemplo, o usuário escolhe o período como deseja ser alertado, se em minutos, horas, dias ou semanas. Note-se que estes valores são apresentados para o usuário transformados em números inteiros, por exemplo, e avaliados em uma estrutura semelhante ao switch já apresentado. Estrutura similar é exibida na interface de smartphones:

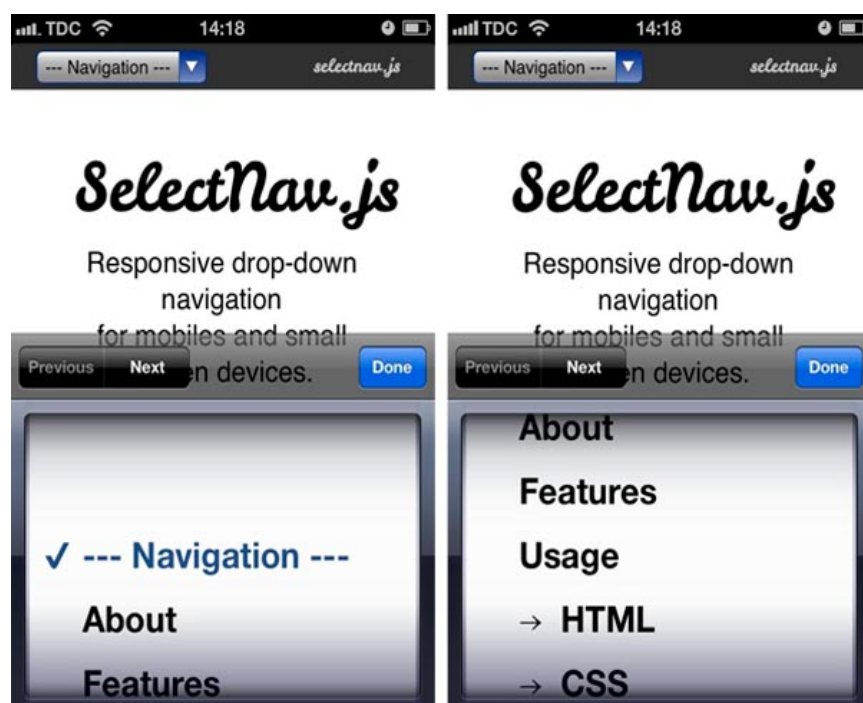


Imagem: Shutterstock.com

DEM QUE EU TE EXPLICO!

Os vídeos a seguir abordam os assuntos mais relevantes do conteúdo que você acabou de estudar.

Estruturas de decisão aninhadas

Estruturas de decisão encadeadas

Estruturas de múltiplas alternativas

VERIFICANDO O APRENDIZADO

1) RETORNANDO AO CASO DA MÉDIA, TEMOS AGORA OUTRA SITUAÇÃO. O ALUNO É CONSIDERADO APROVADO SE A MÉDIA DAS DUAS PROVAS AV1 E AV2 É MAIOR DO QUE 7, PORÉM, SERÁ CONSIDERADO “EM RECUPERAÇÃO” NO CASO DE A MÉDIA ESTAR ENTRE 5 E 7, E ESTARÁ REPROVADO SE A MÉDIA FOR ABAIXO DE 5. A NOTA MÁXIMA DE UM ALUNO É 10. MARQUE A ALTERNATIVA QUE APRESENTA OS TESTES QUE DEVERÃO SER COMPLETADOS NO CÓDIGO ABAIXO.

```
IF (EXPRESSAO1){  
    IF(EXPRESSAO2)  
        PRINTF(“APROVADO”);  
    ELSE  
        PRINTF(“VALOR INCORRETO DE MÉDIA”)  
    ELSE  
        IF(EXPRESSAO3){  
            PRINTF(“EM RECUPERAÇÃO”);  
        }  
    ELSE  
        IF(EXPRESSAO4){  
            PRINTF(“REPROVADO”);  
        }  
    ELSE  
        PRINTF(“VALOR INCORRETO DE MÉDIA”);  
}
```

A) EXPRESSAO1: media>=7

EXPRESSAO2: media<=10

EXPRESSAO3: media>=5

EXPRESSAO4: media>=0

B) EXPRESSAO1: media>=10

EXPRESSAO2: media<=7

EXPRESSAO3: media>=5

EXPRESSAO4: media>=0

C) EXPRESSAO1: media>=7

EXPRESSAO2: media<=10

EXPRESSAO3: media>=0

EXPRESSAO4: media>=5

D) EXPRESSAO1: media>=0

EXPRESSAO2: media<=5

EXPRESSAO3: media>=7

EXPRESSAO4: media>=10

2) CONSIDERE O IF ABAIXO.

INT A=1, B=2 E C;

IF(A>B)

C=3;

ELSE

C=4;

**MARQUE A ALTERNATIVA QUE APRESENTA O OPERADOR TERNÁRIO QUE
SUBSTITUI ESTE IF .**

A) c = (a>b)?3:4;

B) c = (a>b):4?3;

C) c = (a<=b)?3:4;

D) c = (a<=b):4?3;

GABARITO

1) Retornando ao caso da média, temos agora outra situação. O aluno é considerado aprovado se a média das duas provas AV1 e AV2 é maior do que 7, porém, será considerado “em recuperação” no caso de a média estar entre 5 e 7, e estará reprovado se a média for abaixo de 5. A nota máxima de um aluno é 10. Marque a alternativa que apresenta os testes que deverão ser completados no código abaixo.

```
if (EXPRESSAO1){  
    if(EXPRESSAO2)  
        printf(“aprovado”);  
    else  
        printf(“valor incorreto de média”  
    else  
        if(EXPRESSAO3){  
            printf(“em recuperação”);  
        else  
            if(EXPRESSAO4){  
                printf(“reprovado”);  
            else  
                printf(“valor incorreto de média”);
```

A alternativa "A " está correta.

A resposta correta é a letra A. Para ser aprovado, $media \geq 7$, portanto, isso somente poderá ocorrer se a EXPRESSAO1 tiver este valor. A média de uma avaliação necessariamente precisa ser até 10.

Valores de médias maiores não são usuais, pois a média não pode ser maior do que 10. Caso a média não seja ≥ 7 , então o aluno poderá estar de recuperação, ≥ 5 ou reprovado no caso contrário, portanto ≥ 0 . Valores estranhos a esta situação são considerados incorretos.

2) Considere o *if* abaixo.

```
int a=1, b=2 e c;
```

```
if(a>b)  
    c=3;  
else  
    c=4;
```

Marque a alternativa que apresenta o operador ternário que substitui este *if* .

A alternativa **"A "** está correta.

No operador ternário, o símbolo (?) vem antes do símbolo (:), assim, as letras B e D estão incorretas. Neste caso, como $a > b$ é falso, o valor de C deverá ser 4. No exemplo da letra D, se $a = b$, a variável C receberá o valor 4, o que não corresponde ao if apresentado.

CONCLUSÃO

CONSIDERAÇÕES FINAIS

Neste estudo, apresentamos os conceitos de estrutura de decisão suportados pela Linguagem C, que permite que as aplicações desenvolvidas sejam eficientes.

Em um primeiro momento, foram exibidas as estruturas de decisão simples e compostas representadas pelo comando *if* e *if-else*, respectivamente. Posteriormente, vimos o comportamento destas estruturas quando são utilizadas em forma aninhada e encadeada. Adicionalmente, foi apresentado o operador ternário que pode representar um *if* em alguns casos. Para terminar, a estrutura *switch-case*, que é muito utilizada hoje em dia.

Para ouvir um *podcast* sobre o assunto, acesse a versão online deste conteúdo.



REFERÊNCIAS

ADHIKARI, Suman. **Artificial Intelligence, Machine Learning and Deep Learning – Are they same?**

In: Code Heroku. Publicado em: 25 mar 2019.

DAMAS, Luis. **Linguagem C**. 10. ed. Rio de Janeiro: LTC, 2016

SCHILDT, Herbert. **C Completo e Total**. 3. ed. São Paulo: Pearson Education do Brasil, 1996.

EXPLORE+

Para saber mais sobre este tema, sugerimos que você pesquise a linguagem HTML.

As linguagens de programação representam as estruturas de controle que determinarão o comportamento de diversas aplicações. As aplicações que representam o algoritmo que foi implementado, atualmente, revelam o que se encontra no backend de uma aplicação. Normalmente, as opções que são apresentadas aos usuários utilizam outras linguagens, porém não de programação, e sim, visuais, como o HTML.

CONTEUDISTA

Anderson Fernandes Pereira dos Santos

 **CURRÍCULO LATTES**