



5. Estruturas de Decisão

Neste capítulo você aprenderá a:

- Transformar um algoritmo que possua decisão em um programa em *javascript*.
- Utilizar comandos de se-senão no *javascript*.
- Utilizar comandos de seleção múltipla no *javascript*.

5.1 Mudança de Fluxo

Vamos começar a fazer mudança no fluxo dos nossos problemas. A mudança que me refiro é que usualmente todo programa irá executar linha por linha do código-fonte, porém, existem comandos que fazem com que o programa “pule linhas”, ou seja, eles mudam o fluxo linear do programa. Fizemos um algoritmo que possui essa propriedade no Exemplo 2.6, aplicado ao Problema 2.2.3. Onde conforme o resultado da decisão o fluxo seguia uma direção diferente.

Nas representações de fluxograma e pseudocódigo fica evidente que existe uma mudança de fluxo que ocorre com base na condição de um **se** (Exemplo 2.6 e Exemplo 2.9). Veremos como transformá-las em um programa em *javascript*. Sem mais delongas vamos conhecer as seguintes estruturas de decisão: **se**, **se-senão**, **se-senão se-senão** e **caso-selecione**. Cada uma das estruturas será representada com um fluxograma para melhor assimilação dos conceitos e exemplos. Vamos lá!

5.2 Estrutura Se

Toda estrutura de decisão que possui um **se** terá que verificar uma condição. Essa condição, proposição, deverá ser verdadeira para que o escopo (conjunto de instruções) do **se** seja executado. Observe o fluxograma na Figura 5.1, o utilizaremos como base para os exemplos. Mas, primeiro darei um exemplo de uso mais prático.

Você está com sede e está indo para a casa de um amigo. Chegando lá você espera que lhe ofereçam água, se te oferecem água você bebe e mata sua sede, caso não aconteça você continua com sede. Agora vamos associar isso com a nossa estrutura **se**. Caso te

ofereçam água a condição foi satisfeita você entrou na estrutura do **se** e bebeu, porém, senão acontecer você continua na mesma (com sede), o **se** não foi executado. Entendeu!?

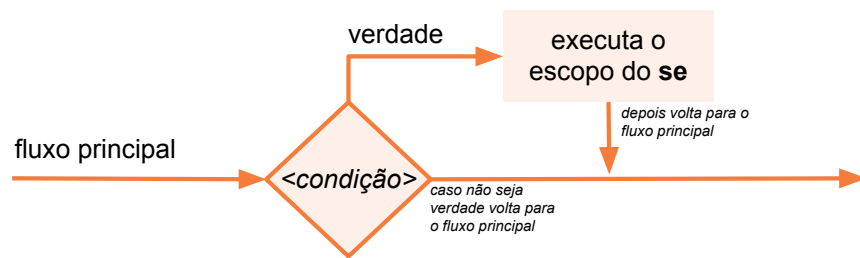


Figura 5.1: Fluxograma da estrutura do **se**. Caso a condição do **se** seja verdadeira o escopo (conjunto de instruções) internas será executado, caso contrário o programa segue o fluxo normal.

Fiz o exemplo da sede em um fluxograma (Figura 5.2). Com base nele faremos o código em *javascript*. Vamos primeiro ver a estrutura **se** no *javascript*:

```

1  if (<condição>) {
2      <instruções>
3  }

```

A palavra-chave que vai identificar essa estrutura é o **if**, após você adicioná-lo entre parênteses, informe a condição. Lembre-se que a condição tem que ser algo que vai retornar **true** ou **false**. Entre chaves você irá colocar todas as instruções que você quiser executar, ou seja, as chaves definem o tal do escopo. Pronto, está aí o nosso **se** no *javascript*. Vamos ao código do exemplo da sede.

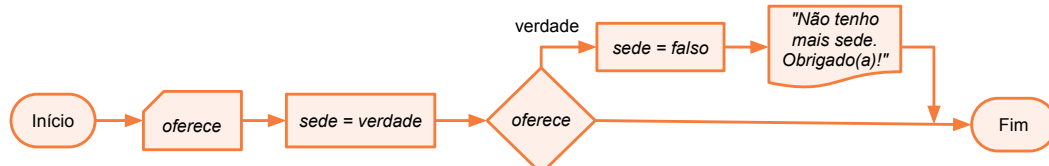


Figura 5.2: Fluxograma da estrutura do **se** no estudo de caso da sede.

■ **Código 5.1** Sede¹ (Fluxograma da Figura 5.2). Para ficar mais dinâmico você irá escrever “Sim”, se seu amigo oferecer água e não escreve nada (dê um enter) caso seu amigo não tiver oferecido.

```

1  var oferece = Boolean(prompt("Seu amigo ofereceu água?"));
2  var sede = true;
3  if (oferece){
4      sede = false;
5      console.log("<strong>Não tenho mais sede. Obrigado(a)!</strong>");
6  }

```



O **Boolean** é um conversor, como o **Number**. Sua ação é transformar um texto em boolean. Caso um texto for vazio ele retorna **false**, caso contrário ele retorna **true**.

¹<<https://replit.com/@DaniloBorges/EstruturaSe-Ex1>>

No Código 5.1, quando executá-lo você não verá nada caso não tiver escrito algo no `console` (dado um `enter`). Por isso, não se assuste. Agora se você digitar algum texto irá aparecer a mensagem de agradecimento.

Convenhamos que tem gente que não tem bom trato com as visitas né... então vamos ter que pedir a água caso não nos ofereçam.

5.3 Estrutura Se-Senão

Essa estrutura também precisa de um **se** com uma condição, porém, caso ela não seja satisfeita podemos fazer outra ação. Sendo assim, ao existir um **senão** garantidamente um dos escopos será executado, o escopo do **se**, ou o escopo do **senão**. Vejamos o fluxograma dessa estrutura na Figura 5.3.

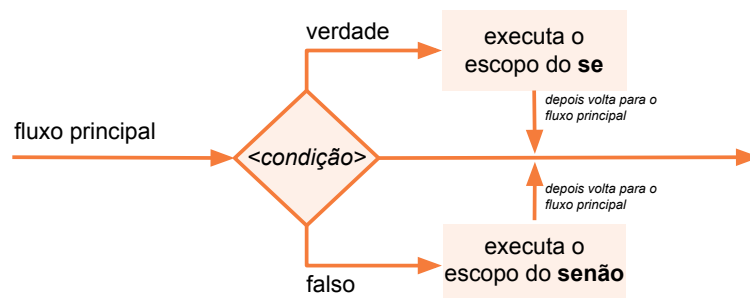


Figura 5.3: Fluxograma da estrutura do **se-senão**. Caso a condição do **se** seja verdade o escopo do **se** será executado, caso contrário o escopo do **senão** será executado. Não importa qual será executado, o fato é que um deles será e depois o fluxo principal é retomado.

Agora podemos fazer o seguinte, se não (**senão**) nos oferecerem água podemos pedir um copo com água. Vamos fazer um código similar ao que o fluxograma da Figura 5.4.

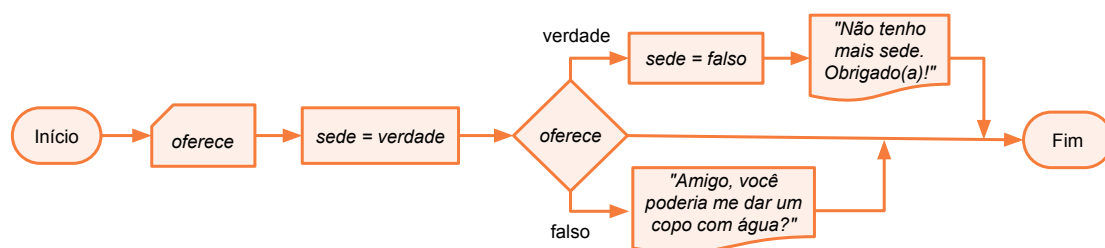


Figura 5.4: Fluxograma da estrutura do **se-senão** no estudo de caso da sede.

A estrutura do **se-senão** começa com a estrutura do **se**, no final do escopo do `if` adicionamos a palavra-chave `else`, que indica o nosso **senão**:

```

1  if (<condição>) {
2      <instruções>
3  }else{
4      <instruções>
5  }

```

Observe que o `else` também possui a definição do escopo, as chaves, todas as instruções que tiverem dentro serão executadas caso a **condição** não for satisfeita. Vamos ver o código do exemplo da sede usando essa nova estrutura:

■ **Código 5.2** Sede com Senão² (Fluxograma da Figura 5.4). Esse código é similar ao Código 5.1, a diferença está na presença do `else`.

```
1 var oferece = Boolean(prompt("Seu amigo ofereceu água?"));
2 var sede = true;
3 if (oferece){
4     sede = false;
5     console.log("Não tenho mais sede. Obrigado(a)!");
6 }else{
7     console.log("Amigo, você poderia me dar um copo com água?")
8 }
```

Agora com este `else`, caso seu amigo não lhe ofereça água você pode pedir. O `else` NÃO existe sem a existência do `if`, note que o colocamos no final do escopo do `if`, depois do encerramento da chave (`...}else`, linha 6). E só podemos ter um único `else` após o `if`.

Agora podemos também fazer o código do Exemplo 2.6, que usa uma estrutura de `se-senão`:

■ **Código 5.3** Raiz da Equação de Primeiro Grau³.

```
1 var a = Number(prompt("Informe o valor de a"));
2 var b = Number(prompt("Informe o valor de b"));
3 if(a==0){
4     console.log("Não existe raiz");
5 }else{
6     var x = -b/a;
7     console.log(`A raiz da equação ${a}x+(${b}) é ${x}`);
8 }
```

❗ O `else` é opcional, você irá utilizá-lo sempre que você ver a necessidade de que alguma coisa deve acontecer se as condições do `if` não serem satisfeitas (verdadeiras).

5.4 Estrutura Se-Senão Se-Senão

Tranquilo até aqui? Espero que sim. Vamos ver agora mais uma possibilidade de mudar o fluxo do nosso programa. Ocorrem casos em que queremos verificar várias situações antes de decidir o que fazer. Vejamos um exemplo disso no contexto universitário.

Usualmente existem critérios para aprovação e reprovação do(a) aluno(a) mediante a sua nota, e mediante essa nota ainda pode ter uma chance de aprovação caso obtenha uma nota boa na avaliação final (AF). Vou explicar como pode ocorrer isso, passo a passo:

1. Calcula-se a média aritmética de três notas do(a) aluno(a);
2. Se a média for maior ou igual a 7, o(a) aluno(a) estará aprovado(a) na disciplina;
3. Senão se a média for maior ou igual a 4 e menor que 7, o(a) aluno(a), terá que fazer a AF da disciplina;
4. Senão o(a) aluno(a), estará reprovado(a) na disciplina.

Observe que apareceu ali o texto “Senão se”, é neste conceito que trabalharemos agora. Vejamos o fluxograma desta estrutura na Figura 5.5. Observa-se que os `se` abaixo só serão executados se o `se` anterior não for verdade, caso qualquer um dos `se` for verdadeiro os

²<<https://replit.com/@DaniloBorges/EstruturaSeSenao-Ex1>>

³<<https://replit.com/@DaniloBorges/EstruturaSeSenao-Ex2>>

posteriores não serão executados. Em resumo, se você quiser verificar várias condições onde só uma delas você quer que seja satisfeita recorra a esta estrutura.

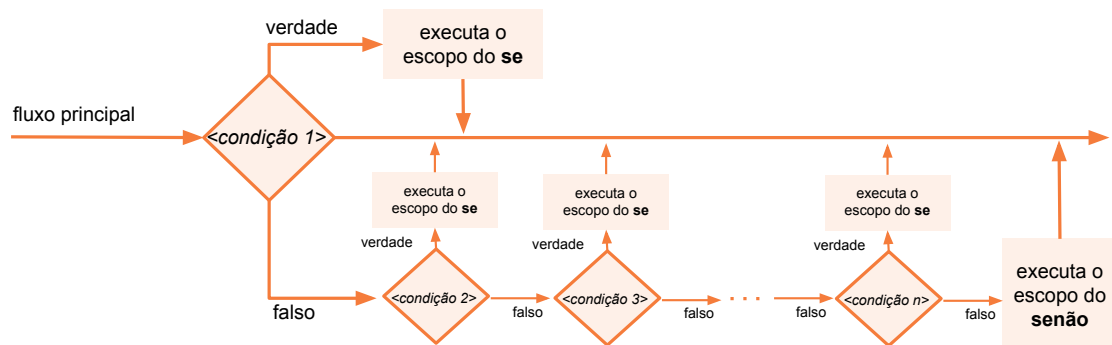


Figura 5.5: Fluxograma da estrutura do **se-se senão-senão**. Caso a condição de algum **se** seja verdade o escopo deste **se** será executado, caso nenhuma condição seja satisfeita o escopo do **senão** será executado.

Vejamos como fica o código em **javascript** da estrutura **se-se senão-senão**:

```

1  if (<condição>) {
2    <instruções>
3  }else if (<condição>){
4    <instruções>
5  }else{
6    <instruções>
7  }

```

Para ilustrar coloquei somente um **senão se**. Você pode colocar quantos **else if** (linha 3) quiser, sempre com o escopo definido (chaves); mas tenha cuidado em relação à posição do **else** sozinho, ele só irá aparecer no final. O **else** como disse é opcional, logo, se for necessário e dependendo do problema, você não precisa colocar o **else**.

Na Figura 5.6, temos o fluxograma do problema da nota. E no Código 5.4 temos seu funcionamento em *javascript*.

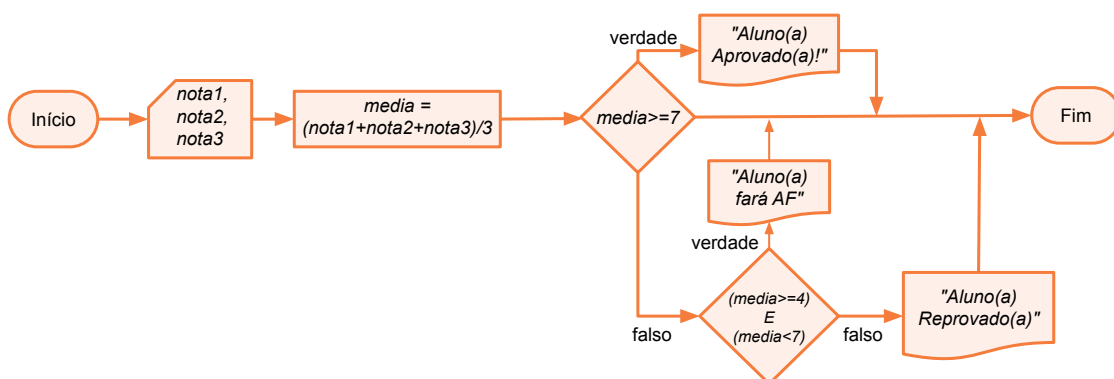


Figura 5.6: Fluxograma da estrutura do **se-senão se-senão** do exemplo da nota.

O Código 5.4 pode ser modificado para outra verificação no comando **else if**. Por exemplo, você poderia alterar a condição para **media < 4** e colocar o texto de impressão que o aluno está reprovado e no **else** colocaria que está de AF. Ou seja, você pode livremente

mudar as condições. Porém tenha cuidado para não colocar condições que são satisfeitas em mais de um `if`, o ideal é que cada condição esteja associada somente a um dos escopos.

■ Código 5.4 Resultado da Média⁴.

```

1  var nota1 = Number(prompt("Insira a primeira nota"));
2  var nota2 = Number(prompt("Insira a segunda nota"));
3  var nota3 = Number(prompt("Insira a terceira nota"));
4  media = (nota1+nota2+nota3)/3;
5  if(media>=7){
6      console.log("Aluno(a) Aprovado(a)");
7  }else if((media>=4) && (media<7)){
8      console.log("Aluno(a) fará AF");
9  }else{
10     console.log("Aluno(a) Reprovado(a)");
11 }

```

Em resumo, sempre que você quiser fazer um teste (**se**) você usará o `if`, se já existir um `if` você terá que colocar um `else if`. Caso queira recorrer a um escopo se nenhuma condição for satisfeita você coloca o `else`. Cada um dos três comandos devem possuir a abertura e fechamento das chaves. Beleza!?

5.5 Estrutura Caso-Seleção

Esta estrutura difere dos nossos comandos de **se**, mas é bem simples. Observe o fluxograma na Figura 5.7. Nesta estrutura é verificada se acontece uma igualdade da variável em uma lista de opções. Então, é verificado se a variável é igual a um determinado valor, x , se for, o escopo será executado; caso nenhum seja executado, opcionalmente, você poderá colocar um escopo final. Esse escopo final é como se fosse o `else` do `if`.

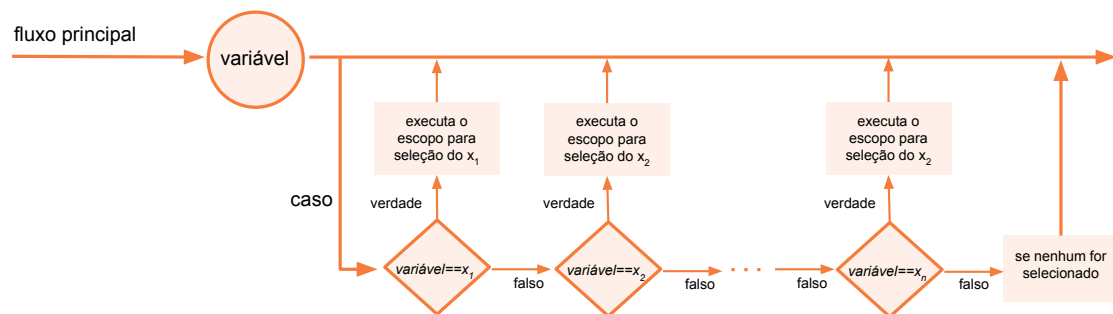


Figura 5.7: Fluxograma da estrutura do **caso-seleção**. Aqui temos uma nova representação de fluxograma o círculo que indica o **caso**.

Para exemplificar vamos trabalhar com o seguinte exemplo: A pessoa vai jogar um jogo chamado *jokenpo*. Ela deverá selecionar uma das seguintes opções: 1, para ser pedra (👊); 2, para ser papel (👐); e 3, para ser tesoura (✂️). O programa deve informar essas opções e depois informar qual opção o usuário selecionou. Caso não tenha colocado um dos três valores o programa deve dizer que não foi informado uma opção válida. Baseado nisso temos o fluxograma deste problema na Figura 5.8.

⁴<<https://replit.com/@DaniloBorges/EstruturaSeSeSenaoSe-Ex1>>

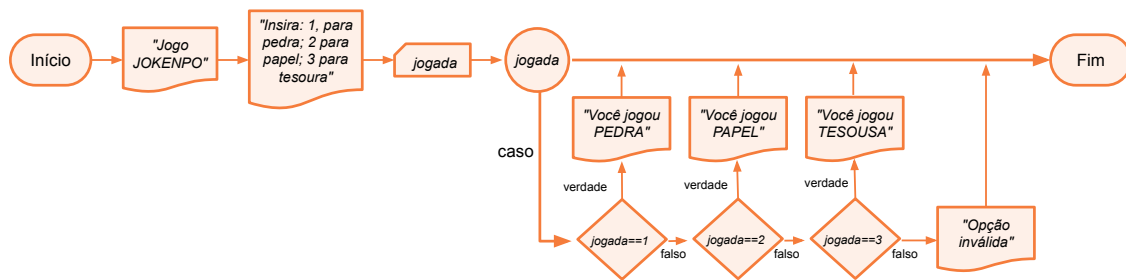


Figura 5.8: Fluxograma da estrutura do **caso-seleção** do exemplo do *jokenpo*.

No fluxograma (Figura 5.8) observe que vamos testar uma variável entre uma lista de opções, selecionamos a opção na qual a igualdade é satisfeita, caso isso não aconteça caímos na última instrução que é imprimir que a opção é inválida. Vejamos agora a estrutura deste comando no *javascript*, chamada de **switch**:

```

1  switch(<variável>){
2    case <valor_1>:{
3      <instruções>
4      break
5    }
6    case <valor_2>:{
7      <instruções>
8      break
9    }
10   default :{
11     <instruções>
12   }
13 }

```

Coloquei aqui somente duas seleções (linhas 2 e 6), mas você pode colocar quantas quiser. Cada escopo da seleção irá definir as instruções que você quer executar e no final você deve colocar o comando **break**. Esse comando é responsável por parar o fluxo o **switch**. Observe que existe uma opção que não tem um **case**, na linha 10 temos o **default**. O **default** será executado sempre que não houver nenhuma seleção válida e sempre ficará por último. Legal né!? Já dá para imaginar muitas aplicações desta estrutura. Vamos ao Código 5.5.

■ Código 5.5 Jogada no Jokenpo⁵.

```

1  console.log("Jogo JOKENPO");
2  var jogada = Number(prompt("Insira: 1, para pedra; 2, para papel; 3,
    para tesoura"));
3  switch(jogada){
4    case 1:{
5      console.log("VOCÊ jogou PEDRA");
6      break;
7    }
8    case 2:{
9      console.log("VOCÊ jogou PAPEL");
10     break;
11   }
12   case 3:{
13     console.log("VOCÊ jogou TESOURA");
14     break;

```

⁵<<https://replit.com/@DaniloBorges/EstruturaCasoSelezione-Ex1>>

```

15     }
16     default:{
17         console.log("Opção inválida");
18     }
19 }

```

O `switch` é um comando que permite com que você possa minimizar o código, visto que com a estrutura de `if`, `else if` e `else` pode-se tranquilamente fazer o mesmo programa, porém, levariam mais linhas de código.

5.6 Resumo

São muitas estruturas e com a prática você irá aprendendo quais utilizar em cada situação. Recomendo que você faça os exercícios e pratique tudo o que vimos até aqui. Coloquei na Tabela 5.1 os comandos em *javascript* com a descrição de uso.

Comando	Descrição
<code>if(<condição>)</code>	Corresponde ao se , você deve colocar em <code><condição></code> uma expressão que retorna um valor booleano, caso seja verdade o escopo do se será executado.
<code>else if(<condição>)</code>	Essa estrutura avalia se uma outra <code><condição></code> é satisfeita. Este comando só pode aparecer depois de um <code>if</code> .
<code>else</code>	Essa estrutura informa se caso nenhum <code>if</code> , ou <code>else if</code> , for satisfeito (suas condições) deverá ser executado seu escopo.
<code>switch(<variável>)</code>	Corresponde a inicialização da estrutura do caso-selecione . Esse comando irá avaliar uma determinada <code><variável></code> dentro de seu escopo constituído por cases .
<code>case <valor></code>	Avalia, dentro do escopo do <code>switch</code> , se a <code><variável></code> é igual a <code><valor></code> . Caso seja verdade seu escopo é executado.
<code>break</code>	Comando que para a execução do <code>switch</code> . Deve ser colocado no final do escopo de cada <code>case</code> do <code>switch</code> .
<code>default</code>	Esse comando é colocado no final do escopo do <code>switch</code> como alternativa caso nenhum <code>case</code> seja satisfeito.

Tabela 5.1: Resumo dos comandos das estruturas de seleção em *javascript*.

5.7 Exercícios

Exercício 5.1 Faça um programa que leia os valores A, B, C e imprima na tela se a soma de A + B é menor que C. Dica: utilize a estrutura **se-senão**.

Exercício 5.2 Faça um algoritmo que leia dois valores inteiros A e B se os valores forem iguais deverá se somar os dois, caso contrário multiplique A por B. Ao final de qualquer um dos cálculos deve-se atribuir o resultado para uma variável C e mostrar seu conteúdo

no console. ■

Exercício 5.3 Faça um algoritmo que leia o nome, o sexo e o estado civil de uma pessoa. Caso estado civil seja "CASADO" ou "CASADA", solicitar o tempo de casado(a) (anos). Dica: utilize a estrutura **se**. ■

Exercício 5.4 Faça um algoritmo para receber um número qualquer e informar na tela se é par ou ímpar. Dica: utilize o operador de módulo e a estrutura **se-senão**. ■

Exercício 5.5 Faça um algoritmo que leia uma variável e some 5 caso seja par ou some 8 caso seja ímpar, imprimir o resultado desta operação. Dica: utilize o operador de módulo e a estrutura **se-senão**. ■

Exercício 5.6 Escreva um algoritmo que leia três valores inteiros e diferentes e mostre-os em ordem decrescente. Dica: utilize a estrutura **se-senão se-senão**. ■

Exercício 5.7 Crie um programa que leia a idade de uma pessoa e informe a sua classe eleitoral:

1. não eleitor (abaixo de 16 anos);
2. eleitor obrigatório (entre a faixa de 18 e menor de 65 anos);
3. eleitor facultativo (de 16 até 18 anos e maior de 65 anos, inclusive).

Dica: utilize a estrutura **se-senão se-senão**. ■

Exercício 5.8 O IMC – Índice de Massa Corporal é um critério da Organização Mundial de Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é $IMC = \text{peso} / (\text{altura})^2$

Elabore um algoritmo que leia o peso e a altura de um adulto e mostre sua condição de acordo com a tabela abaixo.

IMC em adultos	Condição
Abaixo de 18.5	Abaixo do peso
Entre 18.5 e 25	Peso normal
Entre 25 e 30	Acima do peso
Acima de 30	Obeso

■

Exercício 5.9 Elabore um algoritmo que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado.

Código	Condição de Pagamento
1	À vista em dinheiro ou cheque, recebe 10% de desconto
2	À vista no cartão de crédito, recebe 15% de desconto
3	Em duas vezes, preço normal de etiqueta sem juros
4	Em duas vezes, preço normal de etiqueta mais juros de 10%

Exercício 5.10 Criar um programa que leia um número inteiro entre 1 e 7 e escreva o dia da semana correspondente. Caso o usuário digite um número fora desse intervalo, deverá aparecer uma mensagem informando que não existe dia da semana com esse número. ■

Exercício 5.11 Criar um programa que leia um número inteiro entre 1 e 12 e escrever o mês correspondente. Caso o usuário digite um número fora desse intervalo, deverá aparecer uma mensagem informando que não existe mês com este número. ■