



CURSO PROGRAMADOR LIFESTYLE

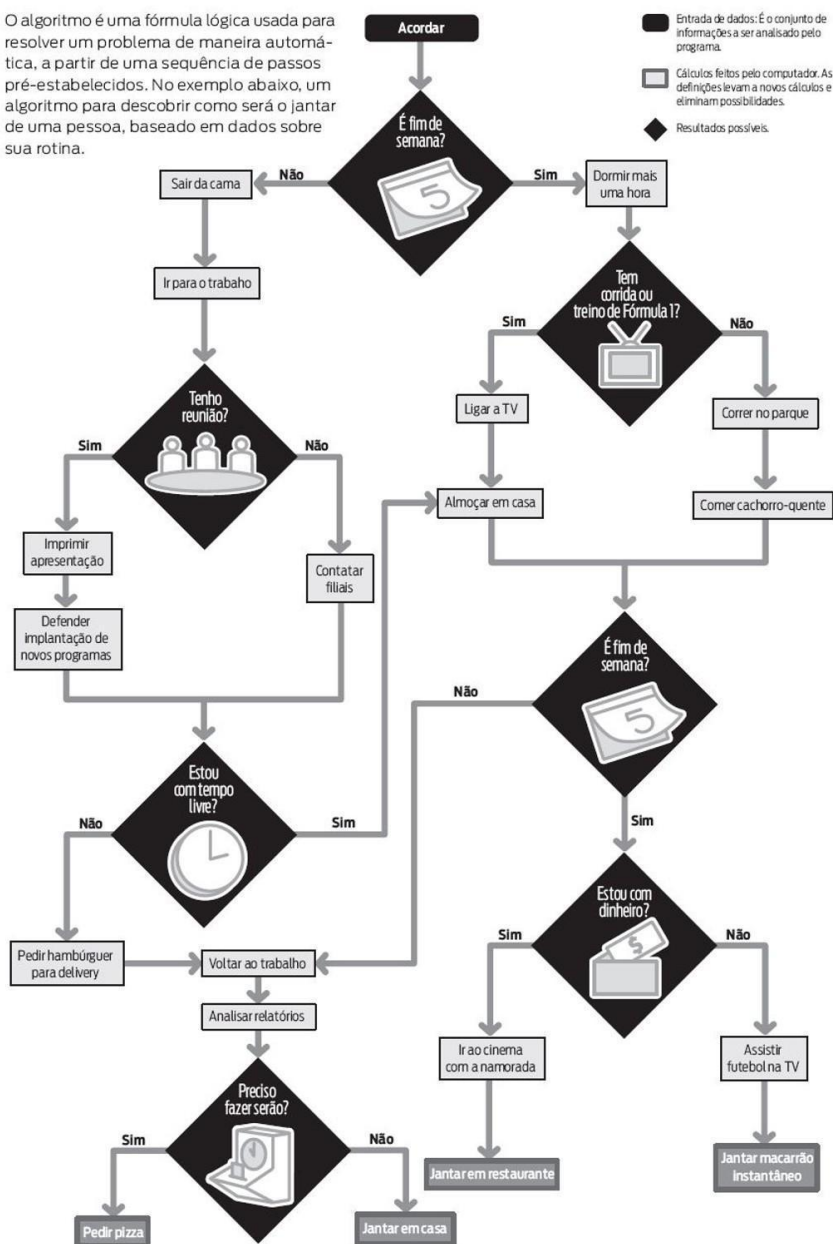
Profs.: Márcio Assis / Jeff

Introdução a Algoritmos

ALGORITMOS, VOCÊ JÁ CONHECE?

Pensando no seu dia-a-dia, suponha que esteja escrevendo seu diário ao final do dia. Observará o quanto os algoritmos estão presentes na sua vida.

O algoritmo é uma fórmula lógica usada para resolver um problema de maneira automática, a partir de uma sequência de passos pré-estabelecidos. No exemplo abaixo, um algoritmo para descobrir como será o jantar de uma pessoa, baseado em dados sobre sua rotina.



Fonte: Gazeta do Povo (2011)

PROBLEMAS E SOLUÇÕES

Vamos supor que temos que pregar um prego em um pedaço de madeira. Para realizar esta tarefa teremos que segurar o prego sobre a madeira e bater com o martelo tantas vezes quantas forem necessárias, até que o prego entre por inteiro.

Uma solução para este problema seria:

1º repetir a seguinte sequência de ações:

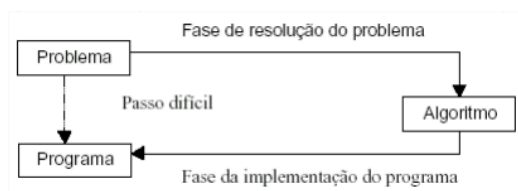
- segurar o prego sobre a madeira com a mão esquerda;
- bater com o martelo no prego, com a mão direita;
- verificar se o prego já está todo dentro da madeira.

Note que nesse exemplo haverá uma repetição de ações até que uma determinada condição esteja satisfeita (o prego esteja dentro da madeira).

Suponha que você precise realizar uma seleção de candidatos para um emprego e há dois requisitos a serem preenchidos. Você deverá contratar os que preencherem os dois requisitos, anotar os dados de quem preencher apenas um dos requisitos e dispensar os que não preencherem nenhum dos dois requisitos. Poderíamos escrever uma solução para este problema da seguinte forma:

- 1º chamar o candidato;
- 2º se preencher os dois requisitos então contratar;
- 3º caso contrário, se preenche um ou outro requisito então anotar seus dados;
- 4º senão dispensá-lo.

O que se pode observar nesta solução é que estamos fazendo uma seleção do candidato que satisfaz as duas condições (requisitos) para o emprego, e também uma segunda seleção (possivelmente para uso futuro) dos candidatos que satisfazem apenas uma das condições. Estas sequências de passos, que resolvem cada um dos exemplos propostos, é o que pode ser chamado de algoritmo para solução do problema. Elas estão a exemplificar as três formas básicas de se definir uma solução para qualquer problema:



SEQUÊNCIA
SELEÇÃO
REPETIÇÃO

Naturalmente que os algoritmos para problemas resolvidos com o auxílio de computadores não serão tão simples e nem apresentados com os exemplos mostrados. A figura ao lado representa a ideia da utilidade dos algoritmos.

ALGORITMOS

Um conjunto finito de regras que provê uma sequência de operações para resolver um tipo de problema específico. Também pode ser definido como uma sequência ordenada, e não ambígua, de passos que levam à solução de um dado problema.

Fazemos uso constantemente de algoritmos em nosso cotidiano: a maneira como uma pessoa toma banho é um algoritmo. Outros algoritmos frequentemente encontrados são:

- Instruções para utilizar um aparelho eletrodoméstico;
- Uma receita para preparo de algum prato;
- Guia de preenchimento para declaração do imposto de renda;
- A maneira como as contas de água, luz e telefone são calculadas mensalmente; etc.

ALGORITMO X PROGRAMA

Um algoritmo é uma sequência lógica de ações a serem executadas para se executar uma determinada tarefa. Um Programa é a formalização de um algoritmo em uma determinada linguagem de programação, segundo suas regras de sintaxe e semântica, de forma a permitir que o computador possa entender a sequência de ações.

LINGUAGEM DE PROGRAMAÇÃO

Linguagem de programação é um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Uma linguagem permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias. C, C++, Java, C# são exemplos de linguagens de programação.

Em Computação, para que o algoritmo possa se tornar um programa executável é necessário escrevê-lo em uma linguagem de programação real, seguindo uma determinada estrutura. Assim, o computador será capaz de “entender” o algoritmo e executá-lo.

ALGORITMOS NÃO ESTRUTURADOS (NARRATIVOS)

Em nossa rotina, executamos algoritmos sem perceber. Quando você levanta pela manhã, quando você sai de casa, quando almoça. Você está sempre executando tarefas enquanto realiza análises de decisões, análises de possibilidades, validando argumentos e diversos outros processos.

Há muitos exemplos de algoritmos, como manuais de instruções de um equipamento, a receita de um prato, etc. Abaixo uma receita de bolo. Ela pode ser considerada um algoritmo? Por quê?

Manuais de instruções também são algoritmos e sempre contêm informações detalhadas sobre o que fazer em cada situação e nos previne de maiores problemas.

Bolo de Chocolate da tia Joana

Ingredientes:

4 xícaras (chá) de farinha de trigo.
2 xícaras (chá) de açúcar cristal.
2 colheres (chá) de achocolatado.
2 colheres (sopa) de fermento em pó.
1 pitada de sal.
3 ovos.
2 xícaras de (chá) de água morna.
1 xícara (chá) de óleo.
Óleo para untar.
Farinha de trigo para polvilhar.

Modo de preparo:

Numa vasilha, *misture* a farinha de trigo, o açúcar cristal, o achocolatado o fermento em pó e 1 pitada de sal. Depois *junte* os ovos, a água morna e o óleo e misture bem. *Unte* uma forma retangular de 25 cm x 37 cm com óleo e polvilhe farinha de trigo. Em seguida *despeje* a massa. Asse em temperatura média (de 170° C a 180° C) por 30 minutos

Situação:

Dirigir até uma lanchonete e almoçar.

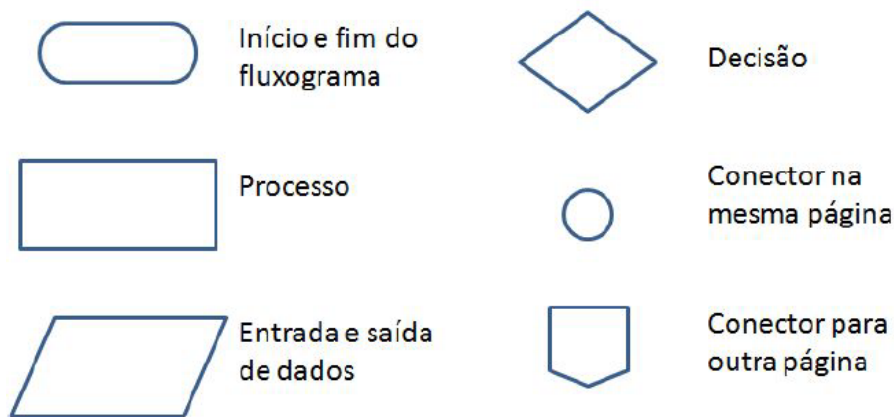
O que é preciso fazer para realizá-la?

- 1º Passo: Saia de casa.
- 2º Passo: Entre no carro.
- 3º Passo: Se a garagem estiver aberta, saia. Senão, abra-a e saia.
- 4º Passo: Dirija até o local desejado.
- 5º Passo: Se chegou, estacione o carro. Senão, continue até chegar e estacionar.
- 6º Passo: Se estacionar, retire a chave da ignição e saia do carro.
- 7º Passo: Tranque o carro e dirija-se à lanchonete.
- 8º Passo: Se a lanchonete estiver aberta, entre e dirija-se ao caixa. Senão, volte para o carro. Procure outra lanchonete aberta, entre e dirija-se ao caixa.
- 9º Passo: Faça o pedido e efetue o pagamento.
- 10º Passo: Aguarde.
- 11º Passo: Se o número de seu pedido for chamado, dirija-se ao caixa e pegue sua refeição. Senão, aguarde ser chamado e pegue-a.
- 12º Passo: Procure um lugar para sentar e coma.
- 13º Passo: Se estiver satisfeito, jogue o resto na lixeira. Senão, faça outro pedido e coma.
- 14º Passo: Saia da lanchonete e dirija-se ao carro.
- 15º Passo: Entre, ponha o cinto de segurança e vá para casa.

Você conseguiu almoçar? Viu como você usa algoritmos na sua vida? Agora pense desde quando você acorda até quando você volta a dormir. Quantas tarefas necessitam de suas decisões? Com certeza muitas. Lógico que não paramos para ver a vida detalhadamente, mas quando percebemos que o que fazemos faz parte de um grande algoritmo de decisões, escolhas, entendemos como funciona um algoritmo computacional.

FLUXOGRAMA

Usam-se símbolos gráficos para representar um algoritmo.



A seguir serão apresentados exemplos de algoritmos em representação gráfica por meio de fluxogramas.

ALGORITMOS ESTRUTURADOS

São aqueles que buscam resolver problemas através do uso de um computador. São criados com base em uma linguagem de programação e podem ser escritos de diversas formas.

Um algoritmo pode ser representado pelo chamado Português Estruturado (pseudocódigo / portugol), que é uma ferramenta que usa combinações de sequências, seleções e repetições. São evitados advérbios e adjetivos, formas verbais diferentes da imperativa, muita pontuação e descrição esparsa. Portugol se trata de uma notação utilizada para escrever programas em uma mescla de português e símbolos comuns das linguagens de programação de alto nível.

Por que programar em Portugol primeiro?

- Simplicidade da linguagem (sintaxe/semântica) apropriada para iniciantes;
- Palavras-chave e comandos (vocabulário) em Português;
- Foco no desenvolvimento da habilidade de resolver problemas usando lógica de programação;
- Semelhança com as linguagens de programação profissionais.

Existem vários ambientes de programação disponíveis: G-Portugol, Visualg, Portugol IDE5, etc. e **Portugol Studio**. Observe esse exemplo bem simples de Algoritmo Estruturado na linguagem Portugol, utilizando a ferramenta Portugol Studio (<http://ite.acad.univali.br/portugol/>).

```
1 programa
2 {
3     funcao inicio()
4     {
5         cadeia time
6         escreva("Digite o nome do time que você torce: ")
7         leia(time)
8         se (time == "Cruzeiro" ou time == "Atlético")
9         {
10             escreva(time, "é um time de MG")
11         }
12         senao
13         {
14             escreva(time, "é um time de outro estado")
15         }
16     }
17 }
```

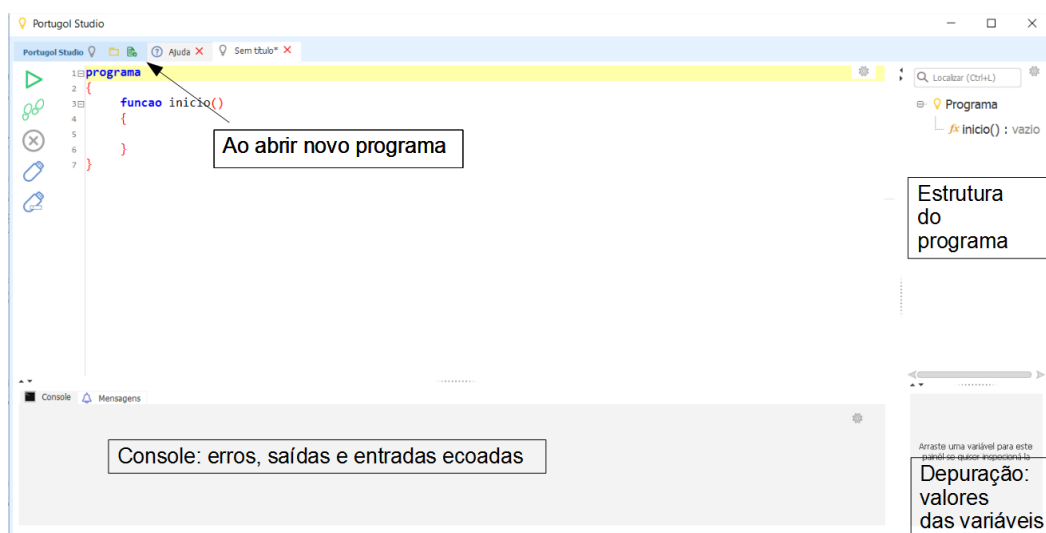
A Portugol Studio é uma IDE (da sigla em Inglês para Integrated Development Environment), ou seja, um ambiente de desenvolvimento de programas, proposto por pesquisadores da Universidade dos Vales do Itajaí (UNIVALI) como um projeto de código fonte aberto, inteiramente escrito em Java, lançado em 2013.

Disponível gratuitamente para download de: <http://lite.acad.univali.br/portugol/>.

A IDE Portugol Studio apresenta as seguintes vantagens:

- foco na aprendizagem ao invés de na produtividade;
- emprega um Portugol com sintaxe mais próxima das linguagens como C, Java e PHP;
- mensagens de erro mais explicativas que em outras IDE's;
- salientador de sintaxe e indicação de erro;
- ambiente simples (ex. cria-se programas e não projetos);
- depurador fácil de utilizar.

A seguir uma imagem ilustrativa do editor de código do Portugol Studio.



Importante destacar que o computador não tem senso próprio, portanto, deve receber instruções explícitas (algoritmos).

Um algoritmo correto deve possuir 4 qualidades:

- Cada passo do algoritmo deve ser uma instrução que possa ser realizada, ou seja, com ações simples e bem definidas;
- A ordem dos passos deve ser precisamente determinada;
- O algoritmo deve ser finito;
- O algoritmo deve ter uma utilidade / um objetivo;
- Deve possuir entradas e saídas.

Para que um algoritmo em **Português Estruturado (pseudocódigo)** seja conciso, ele deve seguir alguns pontos:

- Evitar comandos longos;
- Evitar adjetivos e advérbios;
- Ter comandos legíveis;
- Ter os comandos bem alinhados;
- Possuir comentários para o esclarecimento de comandos;

- Evitar ninhos de SE (muitas possibilidades desnecessárias para uma única situação. Ex.: se chover, eu pego o guarda-chuva; se não chover, eu não pego o guarda-chuva ou se eu pegar o vermelho, ele combinará com a roupa; se eu usar o roxo, não combinará).

METODOLOGIA PARA O DESENVOLVIMENTO DE ALGORITMOS

Para desenvolver um algoritmo que resolva corretamente um problema, começa-se com uma afirmação genérica da solução do problema e prosseguimos até o algoritmo final, aumentando sistematicamente o nível de detalhamento.

Como saber se já temos um nível suficiente de detalhes no algoritmo? Isso depende do agente que irá executar o algoritmo.

- Os computadores têm um conjunto muito limitado de instruções e o algoritmo deve ser expresso nos termos destas instruções.

Abaixo uma sugestão de metodologia para desenvolvimento de algoritmos:

- 1) ler cuidadosamente a especificação do problema e entendê-lo perfeitamente.
- 2) levantar e analisar todas as saídas exigidas na especificação do problema.
- 3) levantar e analisar todas as entradas citadas na especificação do problema.
- 4) verificar se é necessário gerar valores internamente ao algoritmo e levantar as variáveis necessárias e os valores iniciais de cada uma.
- 5) levantar e analisar todas as operações e transformações necessárias para, dadas as entradas e valores gerados internamente, produzir as saídas especificadas.
- 6) testar cada passo do algoritmo, verificando se as transformações intermediárias executadas estão conduzindo aos objetivos desejados. Utilizar, sempre que possível, valores de teste que permitam prever os resultados.
- 7) fazer uma reavaliação geral, elaborando o algoritmo através da integração das partes.

Com essas noções de algoritmos podemos agora começar a programar sem medo. Afinal, todos temos um programador dentro de nós que realiza algoritmos diariamente e com toda a segurança. **Basta acreditar, estudar e fazer acontecer.**

ESTRUTURA DE UM ALGORITMO

O algoritmo é formado por 3 partes básicas:

- **Entrada:** parte do algoritmo onde este recebe os dados necessários para solução do problema;
- **Processamento:** parte onde a entrada é manipulada realizando, por exemplo, os cálculos do problema. É no processamento que está concentrada a parte relativa a lógica do algoritmo;
- **Saída:** meio pelo qual o resultado é exibido.

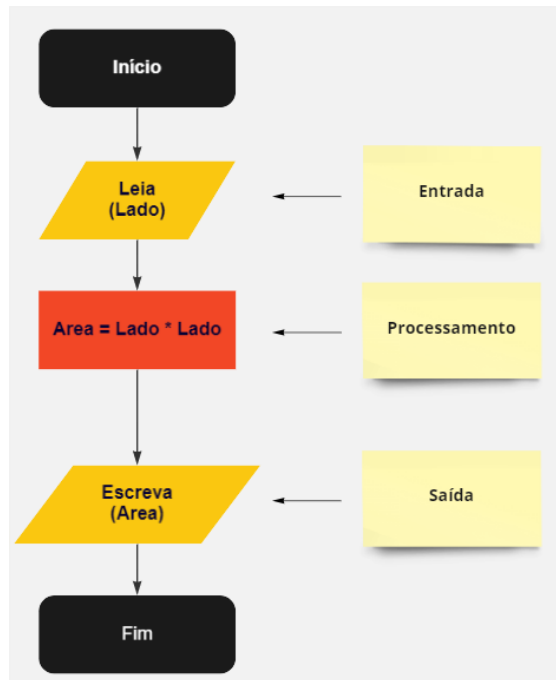
Por exemplo, considere o problema de calcular a **área de um quadrado**. Para isso é necessário saber a medida do lado dessa figura geométrica, logo, a **entrada** para solução do problema é o **lado do quadrado**. Como dito anteriormente, a **saída** corresponde ao resultado que se pretende obter. No caso do nosso problema, a saída é a área do quadrado. Para se chegar neste resultado é preciso aplicar a fórmula da área (**Área = Lado x Lado**). Sendo assim, este cálculo constitui o **processamento** do algoritmo em questão.

Situação: Calcular a área de um quadrado.

O que é preciso fazer para realizá-la?

- 1º Passo: Perguntar qual o lado do quadrado.
- 2º Passo: Memorizar essa informação.
- 3º Passo: Aplicar a fórmula $\text{Área} = \text{Lado}^2$.
- 4º Passo: Responder o valor da Área.

Representando a solução em um fluxograma (diagrama de blocos) teríamos:



O fluxograma acima, ainda que em alto nível, representa um algoritmo, já que retrata uma sequência de passos para calcular a área de um quadrado.

Agora vamos apresentar um exemplo em Portugol, utilizando algoritmo estruturado, resolvendo o mesmo problema.

```
1 programa
2 {
3     funcao inicio()
4     {
5         real lado, area //declaração das variáveis
6         escreva("Digite o lado do quadrado: ") //comando de saída com a mensagem solicitando o lado
7         leia(lado) //comando de entrada
8         area = lado * lado //processamento (execução do cálculo para resolver o problema)
9         escreva("A área do quadrado é ", area) //comando de saída com o resultado
10    }
11 }
```

a) Outro exemplo de algoritmos em fluxograma

Exemplo 2: Fazer um algoritmo que faça o cálculo da conversão do valor de uma temperatura em graus Celsius para graus Fahrenheit. A fórmula de conversão é $F = (9 \cdot C + 160) / 5$, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.

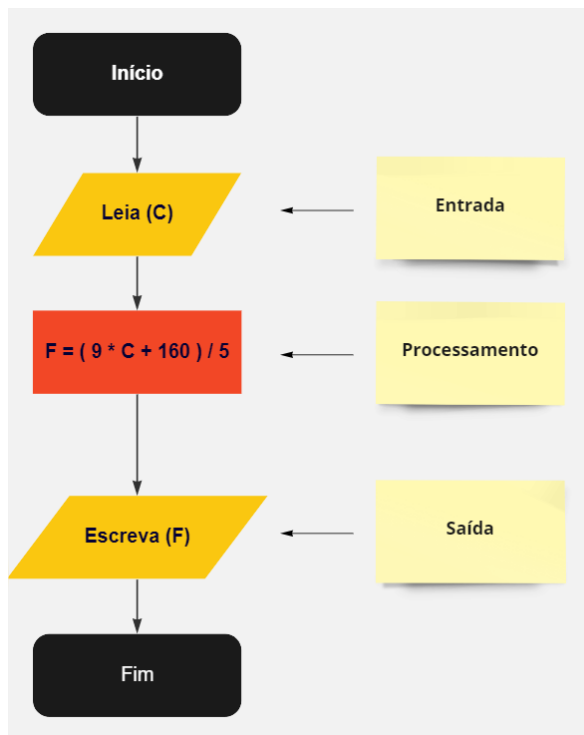
Situação:

Converter o valor de uma temperatura em graus Celsius para graus Fahrenheit.

O que é preciso fazer para realizá-la?

- 1º Passo: Perguntar qual do grau Celsius.
- 2º Passo: Memorizar essa informação.
- 3º Passo: Aplicar a fórmula $F = (9 \cdot C + 160) / 5$.
- 4º Passo: Responder o valor grau Fahrenheit.

Fluxograma:



Abaixo o exemplo em Portugol, utilizando algoritmo estruturado, resolvendo o mesmo problema.

```
1 programa
2 {
3     funcao inicio()
4     {
5         real C, F //declaração das variáveis
6         escreva("Digite o valor do grau Celsius: ")
7         leia(C) //comando de entrada
8         F = (9*C + 160)/5 //processamento (execução do cálculo para resolver o problema)
9         escreva("O grau Fahrenheit é ", F) //comando de saída com o resultado
10    }
11 }
```

CONCEITO ABSTRATO DE VARIÁVEL

Repare que, no exemplo anterior, usamos as letras C e F para representar as temperaturas em graus Celsius e Fahrenheit respectivamente. Em se tratando de algoritmos, C e F são chamadas variáveis. As variáveis são usadas para representar e armazenar as informações que o algoritmo irá manipular.

Além disso, uma informação é classificada como variável quando tem a possibilidade de ser alterado o seu valor em algum instante do tempo. Por exemplo:

- A quantidade de funcionários de uma empresa.
- O peso e idade de uma pessoa.
- A nota de um aluno.
- O salário de um funcionário.

Os algoritmos que iremos construir precisarão receber dados. Por exemplo, o algoritmo (programa) de um caixa eletrônico, recebe dados, certo? E tais dados precisam ser armazenados em algum lugar para posteriormente serem utilizados (sua senha, a identificação da sua conta, etc). Esse armazenamento é feito na memória do computador (que podemos fazer analogia com a memória humana). Uma variável é um “local” nessa memória do computador.

Uma variável possui nome, tipo e tem que ser criada pelo programador quando este estiver construindo um algoritmo que receba e manipule dados (informações).

TIPOS PRIMITIVOS DE DADOS

A informação é a matéria-prima que faz com que seja necessária a existência do computador, pois eles são capazes de manipular e armazenar um grande volume de dados com alto desempenho. Os tipos primitivos de dados são:

Inteiro

Toda e qualquer informação numérica que pertença ao conjunto dos números inteiros (negativa, nula ou positiva). Não possuem parte decimal. A faixa de valores inteiros possíveis é -32767 a 32768.

Por exemplo:

- Hoje é dia 15 de fevereiro.
- Minha idade é 25 anos.
- O professor selecionará 2 alunos para monitoria.

Real

Toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativo, nula ou positiva).

Por exemplo:

- O funcionário receberá R\$ 250,30 por dia.
- A temperatura está próxima do 37,5 graus.

Caracter

Toda e qualquer informação composta por um único caracter alfanumérico (0..9, a .. z, A..Z) e/ou símbolos especiais (por exemplo: *, \$, %, &, #, ?, >,<, !, @). Os números, quando utilizados como caracter, não podem ser utilizados para cálculo.

Por exemplo:

- Estado civil de uma pessoa (C – Casado, S – Solteiro, V – Viúvo, etc).
- Forma de pagamento (A – À vista, P – A prazo, C – Cartão)

Cadeia

Toda e qualquer informação composta por um conjunto de caracteres alfanuméricos (0..9, a .. z, A..Z) e/ou símbolos especiais (por exemplo: *, \$, %, &, #, ?, >,<, !, @). Os números, quando utilizados como caracter, não podem ser utilizados para cálculo.

Por exemplo:

- O nome do funcionário é Antônio Augusto.
- A mensagem de retorno é “Cuidado, o micro está doido”.
- A placa do carro é GXW 0345
- Moro na rua 3.

Lógico

Toda e qualquer informação que pode apenas assumir duas situações possíveis (biestável). Verdadeiro ou Falso.

Por exemplo:

- O funcionário possui dependente? (verdadeiro ou falso)
- A lâmpada pode estar acesa ou apagada?
- O disjuntor está ligado ou desligado?

DECLARAÇÃO DE VARIÁVEL:

Para armazenar corretamente os dados na memória, é necessário a sua correta declaração: tipo primitivo do dado e o seu nome. Devemos obedecer a seguinte sintaxe:

tipo *lista de variáveis*

Exemplo:

inteiro X, Y, Idade, QtdFuncionarios

real Salario, Peso, Dolar, ValorVenda

caracter formapagamento, sexo, estadocivil

cadeia Nome, Endereco, Data

logico TemDependente, MoraOB

REGRAS PARA NOMES (IDENTIFICADORES) DE VARIÁVEIS

Identificador é o nome que podemos dar às variáveis que irão armazenar os dados que precisamos no nosso programa. Para criarmos um identificador, ou seja, para dar nome a uma variável temos que seguir determinadas regras:

- Devem começar por um caracter alfabético (letra);
- Pode ser seguido por mais caracteres alfabéticos e/ou numéricos, ou pelo caracter sublinhado “_”
- Não é permitido o uso de caracteres especiais (@, #, \$, % “, ^, ~, ...);
- Costuma-se usar maiúsculas e minúsculas para separar palavras: "PesoDoCarro";
- Deve ser diferente dos comandos da linguagem;
- Evitar nomes longos, com mais de 31 caracteres;
- Pode conter números a partir do segundo caracter.
- Use nomes coerentes ao que vai ser armazenado.

COMANDOS DE ENTRADA E SAÍDA

No algoritmo é preciso representar a troca de informações que ocorrerá entre o mundo da máquina e o nosso mundo, para isso, devemos utilizar comandos de entrada e saída, sendo que, em nível de algoritmo esses comandos representam apenas a entrada e a saída da informação, independe do dispositivo utilizado (teclado, discos, impressora, monitor,...), mas, sabemos que nas linguagens de programação essa independência não existe, ou seja, nas linguagens de programação temos comandos específicos para cada tipo de unidade de Entrada/Saída.

Comando de Entrada de Dados

Para que possamos obter dados do meio exterior para uso do computador (memória principal), estes virão através dos dispositivos de entrada. Da mesma forma, as informações que são produzidas, serão levadas ao meio externo (um arquivo, uma impressora, uma tela etc.) através de um dispositivo de saída. Para isso, utilizamos dois comandos assim definidos:

Comando **leia()**: lê, do meio externo, a próxima informação disponível para leitura e armazena na(s) variável(eis) discriminada(s) após o comando, entre parênteses. Mais tarde aprenderemos como especificar a leitura de um dado que está armazenado em um arquivo e de que arquivo está sendo lido o dado.

leia(variável_1, variável_2,...)

Ex: leia (v) □ O valor da variável (v) é dado por um dispositivo de entrada e armazenado em memória.

Comando de Saída de Dados

Comando **escreva()**: Imprime (na tela) as mensagens e conteúdo da(s) variável(eis) especificada(s) após o comando, entre parênteses.

Ex: escreva (x) □ O valor atual da variável (x) é informado para um dispositivo de saída.

OPERADORES ARITMÉTICOS

Denominamos expressão aritmética aquela cujos operadores são aritméticos e cujos operandos são constantes e/ou variáveis do tipo numérico (inteiro e/ou real). Os operadores aritméticos são:

| Operador | Descrição | Exemplo |
|----------|----------------------------------------------------------|------------------------------------------------------------------------|
| + | soma | $a + b$ |
| - | subtração | $a - b$ |
| * | multiplicação | $a * b$ |
| / | divisão | a / b |
| / | divisão inteira (se os dois operandos são inteiros) | inteiro x, y ... $x = 5$ $y = 2$ $x / 2$ // será igual a 2 |
| % | módulo da divisão (se os dois operandos são inteiros) | inteiro c = 6 ... $c \% 4$ // será igual a 2 |

Devemos obedecer a precedência dos operadores da seguinte forma:

- Sem uso de parênteses:

1º) / e %

2º) *

3º) + e -

Para mudar a precedência usamos parênteses (e), como nos exemplos da figura ao lado.

```
funcao inicio()
{
    real resultado

    // Neste exemplo, a operação de multiplicação (*) será executada primeiro
    resultado = 5.0 + 4.0 * 2.0
    escreva("Operação: 5 + 4 * 2 = ", resultado)

    // Neste exemplo, a operação de soma (+) será executada primeiro
    resultado = (5.0 + 4.0) * 2.0
    escreva("\nOperação: (5 + 4) * 2 = ", resultado)

    /*
     * Neste exemplo, a operação de divisão (/) será executada primeiro,
     * seguida pela operação de multiplicação (*). Por último, será
     * executada a operação de soma (+)
     */
    resultado = 1.0 + 2.0 / 3.0 * 4.0
    escreva("\nOperação: 1 + 2 / 3 * 4 = ", resultado)

    /*
     * Neste exemplo, a operação de soma (+) será executada primeiro,
     * seguida pela operação de multiplicação (*). Por último, será
     * executada a operação de divisão (/).
     */
    resultado = (1.0 + 2.0) / (3.0 * 4.0)
    escreva("\nOperação: (1 + 2) / (3 * 4) = ", resultado, "\n")
}
```

FUNÇÕES MATEMÁTICAS PRÉ-DEFINIDAS

No primeiro momento, iremos trabalhar apenas com algumas funções matemáticas de Potência e Raiz, que fazem parte da biblioteca **mat** do Portugol Studio.

Para usar as funções matemáticas inclua a biblioteca:

```
programa
{
    inclui biblioteca Matematica --> mat // Inclui a biblioteca Matemática
}
```

E para usar, é necessário colocar o nome da biblioteca **mat**, seguida de ponto (.) e o nome da função.

`mat.potencia(valor, 3.0)` // Calcula o valor elevado ao cubo

`mat.raiz(4, 2.0)` // Calcula a raiz quadrada de 4

EXERCÍCIOS

1. Sabendo-se que uma pessoa recebe um salário fixo por mês e que ele tem uma dívida a pagar. Construa um algoritmo que leia o salário que a pessoa recebe por mês e leia também a dívida que ela possui. Por fim calcule e escreva quanto vai sobrar do salário dela depois de pagar a dívida.
2. Sabendo que uma turma de amigos deseja assistir a um filme e que o ingresso custa R\$14,00, construa um algoritmo que leia o nome do filme que eles irão assistir. Leia também o número de pessoas da turma. Com base nestes dados o seu algoritmo deve calcular quanto a turma gastará para ir ao cinema. Por fim escreva o nome do filme e o total gasto pela turma com os ingressos.
3. Sabendo-se que o preço do passe escolar é 4,50. Um motorista de um ônibus deseja adquirir um programa que leia a quantidade de passageiros que entram no ônibus e que depois calcule e escreva quanto de dinheiro ele terá arrecadado. Construa um algoritmo que resolva o problema do motorista, ou seja leia quantas pessoas entraram no ônibus e depois calcule e escreva quanto foi arrecadado.
4. Sabendo-se que uma pessoa recebe um salário fixo por mês e que além do salário fixo, ela recebe uma gratificação por tempo de serviço. Construa um algoritmo que leia o salário que a pessoa recebe por mês e leia também a gratificação que ela ganhou. Por fim calcule e escreva quanto receberá no total.
5. Após a realização de um censo demográfico em uma dada região, descobriu-se que o número de habitantes cresceu em 25 %. Construa um algoritmo que leia o número habitantes que havia na região antes do censo. Calcule qual é a nova população, conforme aumento citado, e escreva o resultado obtido.
6. Deseja-se cobrir o chão de uma cozinha com pisos de cerâmica. Sabe-se que o formato do chão é de um quadrado e sabe-se também que 1 piso cobre 0,3 metros quadrados. Com base nesses dados calcule e escreva:

- ◆ A área do chão a ser coberto pelo piso;
- ◆ O número de pisos gastos para realizar o serviço desejado.

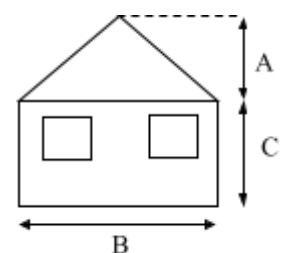
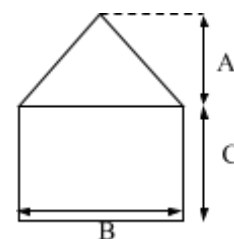
7. Faça um algoritmo que deverá calcular e escrever a área da casinha mostrada ao lado: (OBS.: Os valores de A, B e C serão fornecidos pelo usuário).

8. MODIFICAR o algoritmo anterior para que ele agora calcule o valor da área hachurada da casa (figura abaixo), sabendo que as janelas são quadradas e têm lados iguais a um terço de C.

9. Faça um programa que deverá solicitar ao usuário o raio de uma esfera, calcular e exibir a sua área e o seu volume. Sabe-se que:

$$\text{Área da esfera} = 4 * \pi * \text{raio}^2$$

$$\text{Volume da esfera} = (4 / 3) * \pi * \text{raio}^3$$



10. Um trabalhador recebeu seu salário e o depositou em sua conta bancária. Esse trabalhador emitiu dois cheques e agora deseja saber seu saldo atual. Sabe-se que para cada transação bancária de retirada paga-se uma taxa de 0,38% e o saldo inicial da conta estava zerado. Construa primeiramente um FLUXOGRAMA projetando a solução do problema, que tenha como entrada o valor do salário e valor de cada cheque emitido pelo trabalhador. A saída deverá ser o saldo final na conta, após os dois cheques terem sido descontados. Construa também o algoritmo em Portugol.
11. Sabe-se que são necessários 0,2 litros de tinta para pintar 1m² de parede. Construa um algoritmo que, com base nas medidas de uma parede retangular, escreva a área a ser pintada e a quantidade de tinta necessária para pintar esta parede.
12. Construa um algoritmo que resolva o seguinte problema: Um proprietário de um veículo coletou a informação que seu carro tem um determinado desempenho (consumo de combustível, por exemplo, 11,5 km/litro). Como precisará fazer uma viagem, deseja saber quantos litros de gasolina irá gastar e quanto gastará em dinheiro (R\$). OBS: informações necessárias serão informadas pelo usuário do programa.

13. Faça um algoritmo que receba o valor do salário mínimo, o número de horas trabalhadas, o número de dependentes do funcionário. Calcule e mostre o salário bruto e o salário líquido do funcionário de acordo com as regras a seguir.
- O valor da hora trabalhada é igual a um quinto ($1/5$) do salário mínimo;
 - O salário do mês é igual ao número de horas trabalhadas multiplicadas pelo valor da hora trabalhada;
 - Como auxílio família, para cada dependente acrescentar R\$ 250,00;
 - O salário bruto é igual ao salário do mês, acrescido do auxílio família;
 - Calcular o valor do imposto de renda retido na fonte (IRRF), que será de 27,5%;
 - O salário líquido é igual ao salário bruto menos o IRRF.
14. Construa um algoritmo para resolver o seguinte problema: Obter o valor do salário mínimo e o valor do salário de um funcionário. Calcular e escrever o novo salário deste funcionário sabendo que ele terá um aumento de 7% e quantos salários mínimos este funcionário recebe. Considere o salário do funcionário com o aumento.