Algoritmos

- Seqüência **finita** e ordenada de procedimentos que resolvem um determinado problema

- **Exemplo**: O que preciso fazer para preparar um omelete?
  - **Pegar os ovos na geladeira;**
  - **Bater os ovos;**
  - **Escolher o recheio;**
  - **Pegar a frigideira;**
  - **Pegar o óleo;**
  - **Colocar o óleo na frigideira;**
  - **Ligar o fogão;**
  - **Colocar a frigideira com óleo no fogão;**
  - **Despejar os ovos batidos na frigideira;**
  - **Colocar o recheio sobre os ovos;**
  - **Esperar fritar;**
  - **Retirar do fogo;**
  - **Colocar sobre um prato.**

Sintaxe e Semântica

- **Sintaxe** é o conjunto de regras que devem ser seguidas para a escrita de um algoritmo. Tem relação direta com a **forma**;

- **Semântica** refere-se à ação que é executada pelo computador com um determinado comando. Tem relação com o **conteúdo**;

- Diferenciar sintaxe e semântica é importante!

- **Pergunta**: por que não utilizar diretamente o português para escrever programas?
  - Porque o português é ambíguo (duplo sentido), ou seja, sua semântica não é precisa;
  - Porque a sintaxe do português é complicada demais;

- **Solução:** utilizar uma linguagem com sintaxe mais simples e semântica precisa (português estruturado).

Tipos de dados

- **Dados:** informações em estado primitivo que servem de base para a formulação de algoritmos que, por sua vez, geram informação útil para o usuário;
- Como a natureza do que é armazenado em memória muda, os dados possuem um **tipo** identificando que valores podem ser armazenados;

- **Inteiro:** armazena valores do conjunto dos números inteiros;
  - **Valores numéricos:** 18; 32; -20; 5032.
  - **Exemplos de uso**:
    - Número de funcionários de uma empresa;
    - Quantidade de computadores em estoque;

- **Real:** armazena valores do conjunto dos números reais;
  - **Valores numéricos**: 1,32; 20; 4,56; -7,89.
  - **Exemplos de uso**:
    - Saldo bancário;
    - Salário de um funcionário.

Mais tipos de dados

- **Lógico:** armazena verdadeiro ou falso (de acordo com a álgebra de Boole);
  - **Valores possíveis**: verdadeiro ou falso;
  - **Exemplos de uso**:
    - Estado de funcionamento de uma TV: ligada ou desligada;
    - Condição de um boleto bancário: pago ou não pago;

- **Caractere:** armazena um caractere na memória;
  - **Valores**: letras (A a Z), dígitos (0 a 9); outros símbolos (#; $; %);
  - **Exemplos de uso**:
    - Primeira letra do nome de uma pessoa;
    - Símbolo de uma operação aritmética (+ - / *);
- **Cadeia de caracteres:** representa um conjunto de caracteres a ser armazenado na memória;
  - **Valores:** ‘Carlos de Salles’; ‘455 9098’;
  - **Exemplos de uso:**
    - Nome completo de um cliente;
    - Telefone de uma empresa.

Operadores aritméticos

Símbolo	*Operação*
+	Soma de 2 números
-	Subtração de 2 números
/	Divisão de 2 números
*	Multiplicação de 2 números
#	Divisão inteira de 2 números
%	Resto da divisão de 2 números

Operadores relacionais

Operador	*Relação*
=	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
<>	Diferente de

Operadores lógicos ou booleanos

Operador	*Relação*
E	**Conjunção**
OU	**Disjunção**
NÃO	**Negação**

Tabela-verdade do operador E

Operando 1	Operando 2	Resultado da Conjunção
FALSO	FALSO	FALSO
FALSO	VERDADEIRO	FALSO
VERDADEIRO	FALSO	FALSO
VERDADEIRO	VERDADEIRO	VERDADEIRO

Tabela-verdade do operador OU

Operando 1	Operando 2	Resultado da Disjunção
FALSO	FALSO	FALSO
FALSO	VERDADEIRO	VERDADEIRO
VERDADEIRO	FALSO	VERDADEIRO
VERDADEIRO	VERDADEIRO	VERDADEIRO

Tabela-verdade do operador NÃO

Operando	Resultado da Negação
FALSO	VERDADEIRO
VERDADEIRO	FALSO

Funções

- **Recebem um ou mais parâmetros e retornam um valor resultante, ou seja, são muito parecidas com funções matemáticas;**

- **Sintaxe de funções:**

  – <nome> (<parâmetro 1>, <parâmetro 2>, ...) ;

- **Funções pré-definidas de português estruturado:**

Nome da função	Semântica
QUAD (x)	Quadrado de X
RAIZ (x)	Raiz quadrada de X
TRUNC (x)	Valor inteiro de X sem parte decimal
ARRED (x)	Valor inteiro mais próximo de X
ABS (x)	O valor de X sem sinal

- **Novas funções podem ser definidas pelo programador.**

Precedência em Expressões

- **A avaliação de expressões deve ser feita de forma semelhante a expressões algébricas;**
- **Ordem de precedência de operadores:**
  - 1 – Parênteses mais internos
  - 2 – Funções
  - 3 – Operadores aritméticos
    - 3.1 - Multiplicativos primeiro;
    - 3.2 - Aditivos depois;
  - 4 – Operadores relacionais
  - 5 – Operadores lógicos na ordem:
    - NÃO
    - E
    - OU

- **Exemplos de expressões algorítmicas:**

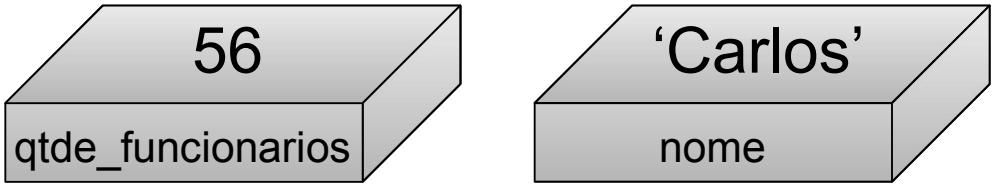
$$\frac{4+4}{2*2}+3^2 \Rightarrow (4+4)/(2*2)+QUAD(3)$$

$$\frac{3*2+1}{9+5}+\sqrt{4} \Rightarrow (3*2+1)/(9+5)+RAIZ(4)$$

$$3 \le x < 5 \vee x \le 4 \Rightarrow (x >= 3 \ \overline{E} \ x < 5) \ \overline{OU} \ x <= 4$$

Variáveis

- Área de memória para o armazenamento de dados;



- Variáveis em algoritmos são sutilmente diferentes do mesmo conceito em matemática. O nome é dado porque aqui seu valor pode variar com o tempo;

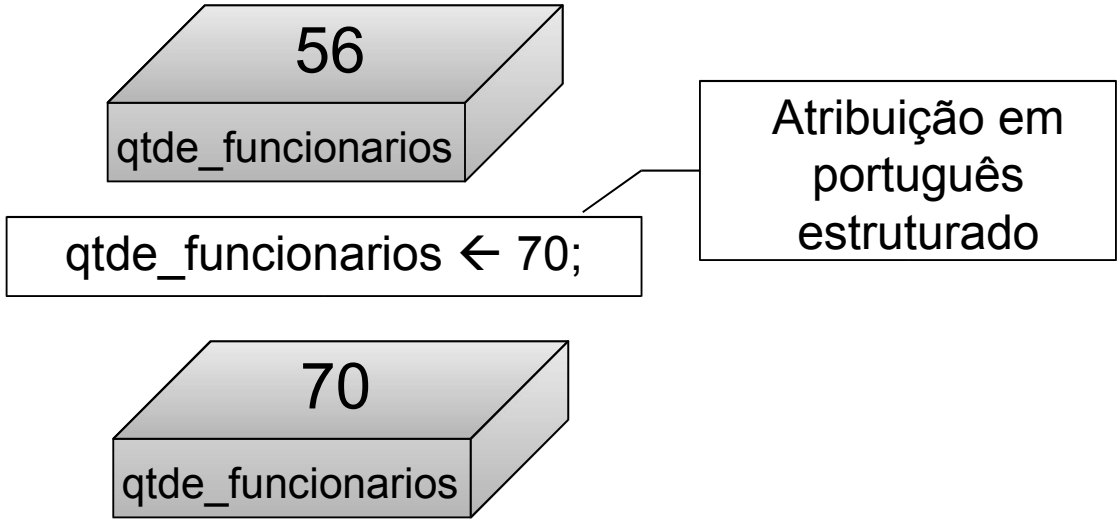
Nomenclatura de variáveis

- Os nomes de variáveis devem começar com uma letra, seguida de outras letras, dígitos, ou sublinhado ("_");
- Toda variável possui um **tipo** associado;
- Os nomes de variáveis devem ser auto-explicativos;
- Exemplos:
  - qtde_funcionarios
  - idade
  - h, gr, qtde **(nomes ruins)**
- Historicamente, as letras **i, j, k** são usadas para contadores

Declaração de variável

- A declaração de uma variável serve para identificá-la (dar um nome) e apresentar seu tipo;
- Sintaxe de uma declaração:

  <variável 1>, <variável 2>, ... :<tipo>
- **Atribuindo valores a variáveis**
  - Uma variável sempre armazena apenas um único valor;



  - Quando uma atribuição é feita, o valor anterior da variável é substituído pelo novo valor informado na atribuição;
  - Atribuições de tipos diferentes são inválidas. Exemplo de atribuição inválida(cadeia a um inteiro):

```
qtde : inteiro;
qtde ← 'Carlos';
```

Comando de entrada de dados

- Sintaxe do comando:
  - **ler** <variável1> , <variável2> , ...

- O comando **ler** é a forma mais rudimentar de entrada de dados por parte do usuário;

- Em português estruturado a entrada de dados é feita exclusivamente por meio desse comando;
- Exemplos:

```
clientes :
inteiro;
ler clientes;
```

  - pede para o usuário informar o número de clientes (um valor inteiro) e armazena o valor digitado na variável clientes;

```
nome_cliente : cadeia;
idade_cliente : inteiro;
ler nome_cliente,
idade_cliente;
```

  - pede para o usuário informar uma cadeia de caracteres e a armazena na variável nome_cliente, depois requisita um número inteiro e armazena na variável idade_cliente.

Comando de saída de dados

- Sintaxe do comando:
  - **escrever** <variável ou expressão ou mensagem>, ...

- O comando **escrever** é a forma mais rudimentar de saída de dados para o usuário;
- Em português estruturado a saída de dados é feita exclusivamente por meio desse comando;
- Exemplos:
  - o resultado desse programa

```
escrever 'Olá. Boa tarde!';
```

  - é a exibição desse texto na tela:

Olá. Boa tarde!

  - O resultado desse programa

```
nome_cliente : cadeia;
idade_cliente : inteiro;
nome_cliente ← 'João';
idade_cliente ← 20;
escrever 'O cliente ',
nome_cliente, ' tem ',
idade_cliente,' anos de idade.';
```

  - é a exibição desse texto na tela:

O cliente João tem 20 anos de idade.

Sintaxe geral de um algoritmo

```
Tipos
    <tiponovo1> = <tipo1>;
    <tiponovo2> = <tipo2>;
Variáveis
    <variável1>, <variável2>, ... : <tipo>;
    <variável3>, <variável4>, ... : <tipo>
//Isso é um comentário
//Esse trecho será ignorado pelo compilador
Início
    <comando1>;
    <comando2>;
    <comando3>;
Fim.

Módulo <nomemódulo1>;
    [ [
    ...
    ] ];

Módulo <nomemódulo2>;
    [ [
    ...
    ] ];

Módulo ...
```

Problema 1 – Equações de 2º grau

- **Uma equação de segundo grau tem o seguinte formato:**

$$ax^2 + bx + c = 0, a \neq 0 \wedge a, b, c \in R$$

- **Com base nisso, escreva um programa que lê os valores a,b,c e exibe o resultado da equação de 2º grau.**
- **Solução:**

```
Variáveis
   a, b, c, x1, x2 : real;
Início
   escrever 'Informe o valor de a:';
   ler a;
   escrever 'Informe o valor de b:';
   ler b;
   escrever 'Informe o valor de c:';
   ler c;
   x1 ← (-b + RAIZ(b*b - 4*a*c))/(2*a);
   x2 ← (-b - RAIZ(b*b - 4*a*c))/(2*a);
   escrever 'X1 = ', x1;
   escrever 'X2 = ', x2;
Fim.
```

Problema 2 – Área de um retângulo

- **A área de um retângulo é definida por:**

$$base * altura$$

- **Faça um programa que receba do usuário a base e altura de um retângulo e exiba como resposta a área do retângulo.**
- **Solução:**

```
Variáveis
    base, altura, area : real;
Início
   escrever 'Informe a base do retângulo:';
   ler base;
   escrever 'Informe a altura do retângulo:';
   ler altura;
   area ← base*altura;
   escrever 'Area = ', area;
Fim.
```

Problema 3 – Rendimentos

- **No Brasil, uma aplicação financeira paga 20% de imposto de renda sobre os rendimentos brutos.**
- **Faça um programa que leia do usuário seu saldo em uma aplicação financeira e os rendimentos em um mês da mesma. Finalmente, exiba como resposta o saldo atualizado da aplicação excluindo o imposto de renda.**

- **Solução:**

```
Variáveis
    saldo_atual, saldo_novo, rendimentos : real;
Início
    escrever 'Informe o saldo do fundo:';
    ler saldo_atual;
    escrever 'Informe os rendimentos:';
    ler rendimentos;
    saldo_novo ← saldo_atual *
       (rendimentos*0.8);
    escrever 'Novo saldo = ', saldo_novo;
Fim.
```

Problema 4 – Qual a senha?

- **Para entrar num sistema de segurança, o espião precisa informar sua senha de acesso. A senha é 'abre-te sesamo'.**
- **Faça um programa que pede ao usuário para informar a senha e exibe VERDADEIRO caso a senha seja correta ou FALSO em caso contrário.**
- **Solução:**

```
Variáveis
    senha : cadeia;
    senha_correta : lógico;
Início
    escrever 'Qual a senha?';
    ler senha;
    senha_correta ← senha='abre-te sesamo';
    escrever senha_correta;
Fim.
```

Problema 5 – Média em uma disciplina

- **A média final em uma disciplina é calculada da seguinte forma:**

$$\frac{nota1 + nota2 + nota3}{3}$$

- **Faça um programa que leia as três notas de um aluno em uma disciplina e exiba sua média. Além disso, o programa exibe VERDADEIRO se o aluno tiver média sete ou mais ou FALSO em caso contrário.**
- **Solução:**

```
Variáveis
    nota1, nota2, nota3, media : real;
    aprovado : lógico;
Início
    escrever 'Informe a nota 1:';
    ler nota1;
    escrever 'Informe a nota 2:';
    ler nota2;
    escrever 'Informe a nota 3:';
    ler nota3;
    media ← (nota1+nota2+nota3)/3;
    aprovado ← media>=7;
    escrever 'Media = ', media;
    escrever 'Aprovado = ', aprovado;
Fim.
```

Problema 6 – Pay per view

- **Em um sistema de TV por assinatura, alguns canais funcionam no sistema *pay per view* e outros não.**
- **Os canais *pay per view* são os múltiplos de 11 e os canais 56 a 62. Todos os demais canais são livres, ou seja, estão inclusos na assinatura mensal.**
- **Faça um programa que leia do usuário um número de canal e exiba na tela a mensagem 'Canal livre FALSO' se o canal for *pay per view* ou 'Canal livre VERDADEIRO' em caso contrário.**

- **Solução:**

```
Variáveis
    canal : inteiro;
    pay_per_view, livre : lógico;
Início
    escrever 'Informe o canal:';
    ler canal;
    pay_per_view ← (canal%11=0) OU
      (canal>=56 E canal<=62);
    livre = NÃO pay_per_view;
    escrever 'Canal livre ', livre;
Fim.
```

Etapas da Construção de um algoritmo

- **Entender o problema;**
- **Identificar as saídas;**
- **Identificar as entradas do programa;**
- **Analisar o processamento;**
- **Manter separadas as etapas de entrada, processamento e saída;**
- **Definir a seqüência de comandos;**
- **Se necessário, realizar a verificação manual de algoritmos (método chinês).**

Modularização de algoritmos

- **Equivale a dividir um problema maior em vários subproblemas, mais simples de serem resolvidos;**
- **Método "dividir para conquistar";**
- **Exemplo: Problema 1 – Equações de 2º Grau**

```
Variáveis
    a, b, c, x1, x2 : real;
Início
    ler_abc;
    calcular_raizes;
    escrever_raizes;
Fim.
Módulo ler_abc;
    [[
    escrever 'Informe o valor de a:';
    ler a;
    escrever 'Informe o valor de b:';
    ler b;
    escrever 'Informe o valor de c:';
    ler c;
    ]];
Módulo calcular_raizes;
    [[
    x1 ← (-b + RAIZ(b*b - 4*a*c))/(2*a);
    x2 ← (-b - RAIZ(b*b - 4*a*c))/(2*a);
    ]];
Módulo escrever raizes;
    [[
    escrever 'X1 = ', x1;
    escrever 'X2 = ', x2;
    ]];
```