

Princípios de Programação

Professor Ricardo Demattê

IFSP – Unidade Salto

Atividade Inicial



Cadastro no Google Classroom

- 1) Acessar o link do **Google Classroom**:
<https://classroom.google.com/>
- 2) Usar sua conta no Gmail ou do **Google** para realizar o cadastro no Google Classroom
- 3) Inserir o código da turma fornecido pelo professor para realizar a matrícula no Ambiente Virtual de Aprendizagem (AVA).



LightBot

- 1) Acessar o link do **LightBot** pelo **computador** ou **celular**:
 - <https://lightbot.com/hour-of-code.html>
 - <https://www.gameflare.com/online-game/light-bot/>
- 2) Aprenda a usar a sua lógica para resolver os problemas e programar o robô para acender todos os quadrados azuis.
- 3) Ao final do jogo será emitido um certificado de conclusão. Insira seu nome e sobrenome no certificado, faça uma cópia e envie ao professor através do Google Classroom.



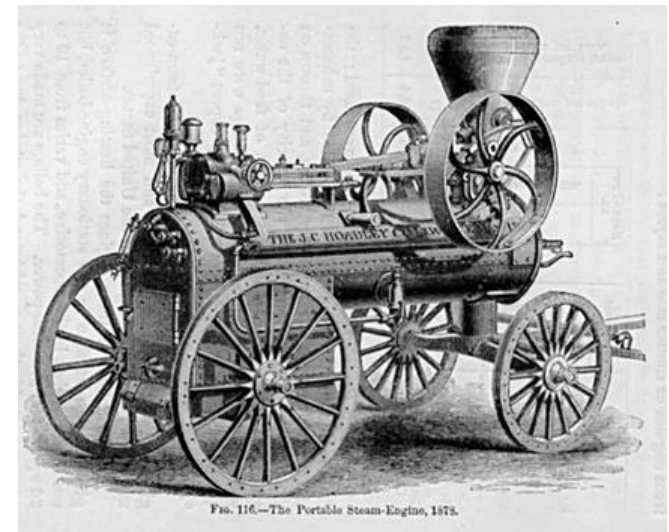
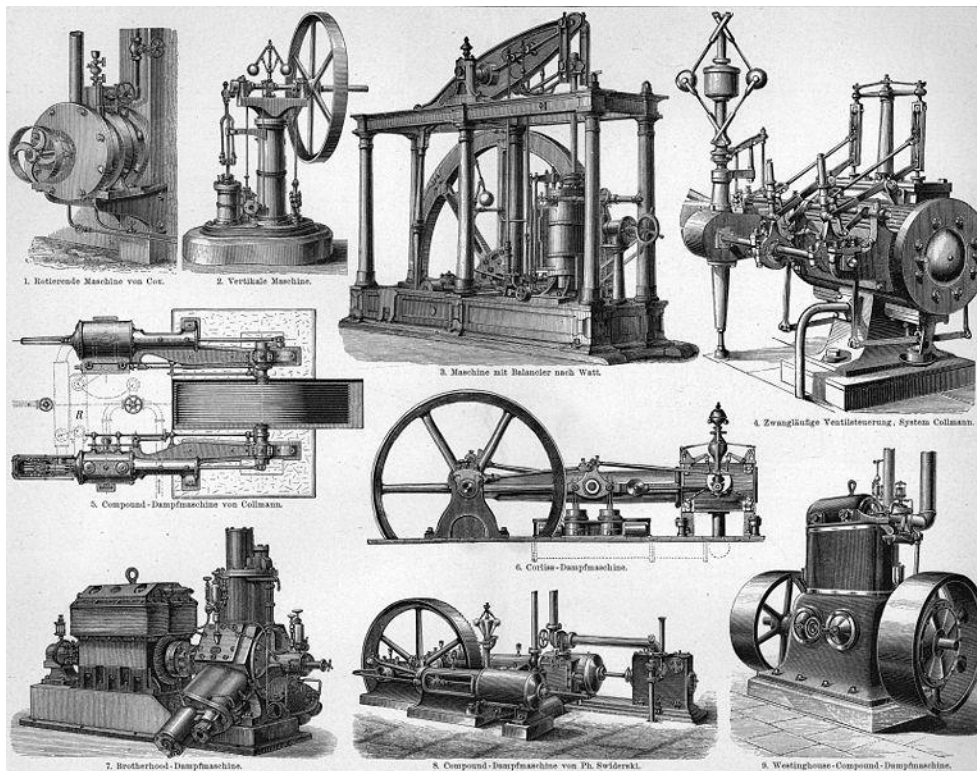
Aula 1 – História da Computação



Revoluções Industriais

1) A Primeira Revolução Industrial – 1760 a 1850

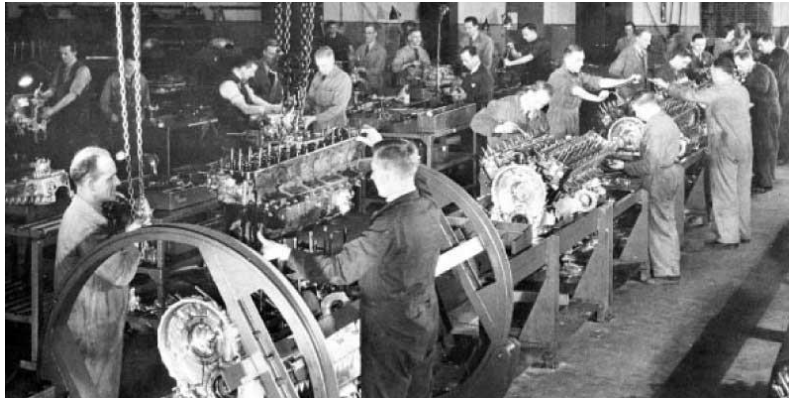
- Capitalismo e a burguesia
- Surgimento das primeiras indústrias e mecanização
- Ferro, carvão e máquinas a vapor



Revoluções Industriais

2) A Segunda Revolução Industrial – 1850 a 1945

- Expansão do capitalismo
- Aprimoramento de técnicas, máquinas e meios de produção
- Aço, máquinas elétricas, petróleo, química e comunicações



Revoluções Industriais

3) A Terceira Revolução Industrial – 1950 - hoje

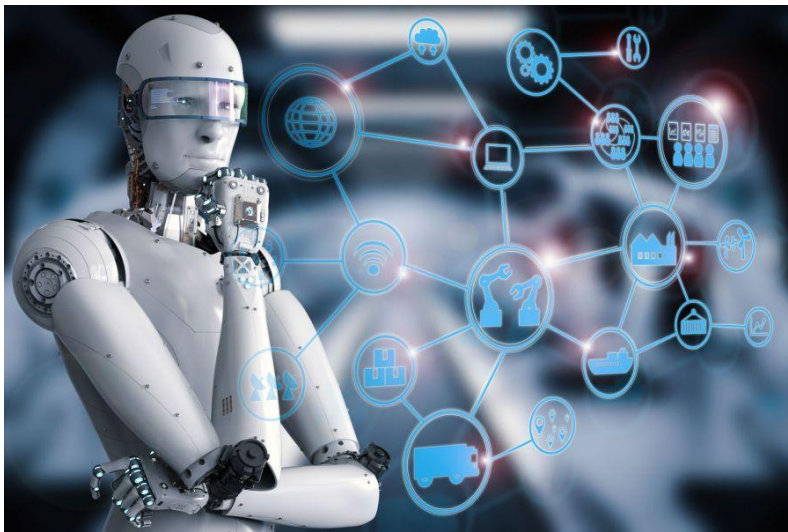
- Empresas multinacionais e globalização
- Avanços tecnológicos e científicos na indústria
- Energia nuclear, automação industrial, robotização, computação



Revoluções Industriais

4) A Quarta Revolução Industrial – hoje - futuro

- Inteligência artificial, robótica, internet, big data, Internet das coisas (IoT), sensores inteligentes, nanotecnologia, biotecnologia
- Operações integradas, menor custo, otimização contínua
- Matriz energética renovável



Revoluções Industriais

Quarta Revolução Industrial

A inteligência está transformando a experiência do cliente



História da Computação

- 1) Assistir **4 filmes**
- 2) Fazer um **resumo de 20 linhas de cada um dos filmes**
- 3) Utilizar o **modelo** disponibilizado pelo professor (Word)
- 4) O trabalho deve ser feito em 2 folhas: **capa** e **texto** com o resumo
- 5) Salvar como **PDF**
- 6) Postar o resumo até a data solicitada.



História da Computação

- **Filme 1 :“O Jogo da Imitação”**
- Relata o início dos computadores durante a Segunda Guerra Mundial
- História do primeiro computador moderno (Alan Turing)



História da Computação

- **Filme 2 : “Piratas do Vale do Silício”**
- Relata o início dos computadores pessoais, os “PCs”
- Histórias da Microsoft (Bill Gates) e Apple (Steve Jobs)



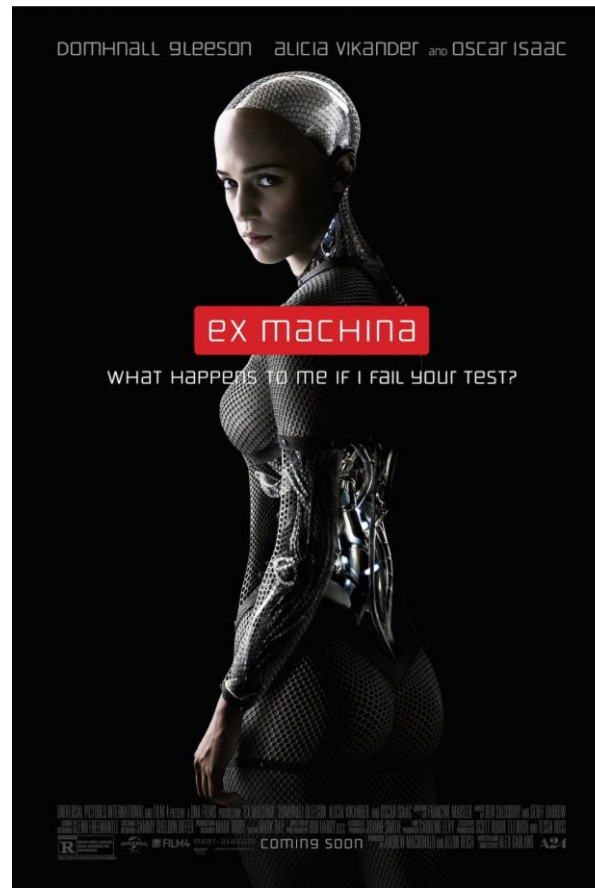
História da Computação

- **Filme 3: “A Rede Social”**
- Relata o início das redes sociais no mundo
- História do Facebook (Mark Zuckerberg)

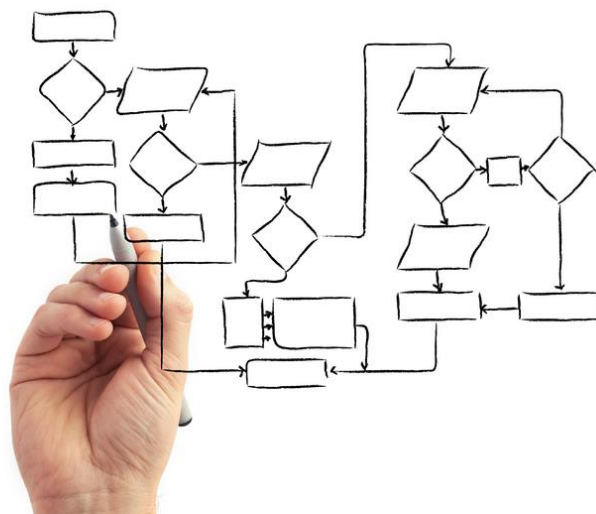


História da Computação

- **Filme 4: “Ex-Machina”**
- Relata um possível futuro com a Inteligência Artificial (IA) e robótica avançada



Aula 2 – Fluxogramas



Fluxo de Processos

- Um processo é uma **sequência contínua de fatos** ou operações que apresentam certa unidade ou que se reproduzem com certa regularidade.
- O processo deve ter ser **eficaz, eficiente, lógico e completo**.
- A **eficácia** refere-se a fazer a tarefa certa, completar atividades e alcançar metas.
- A **eficiência** se refere a fazer as coisas de forma otimizada, de maneira mais rápida ou com menos gastos.
- A **lógica** serve para se pensar corretamente ou racionalmente, utilizando a razão, argumentação ou fala.
- Um processo deve ser **completo**, que não falta parte ou elemento algum dos que o constituem ou que deve ter.

Fluxo de Processos

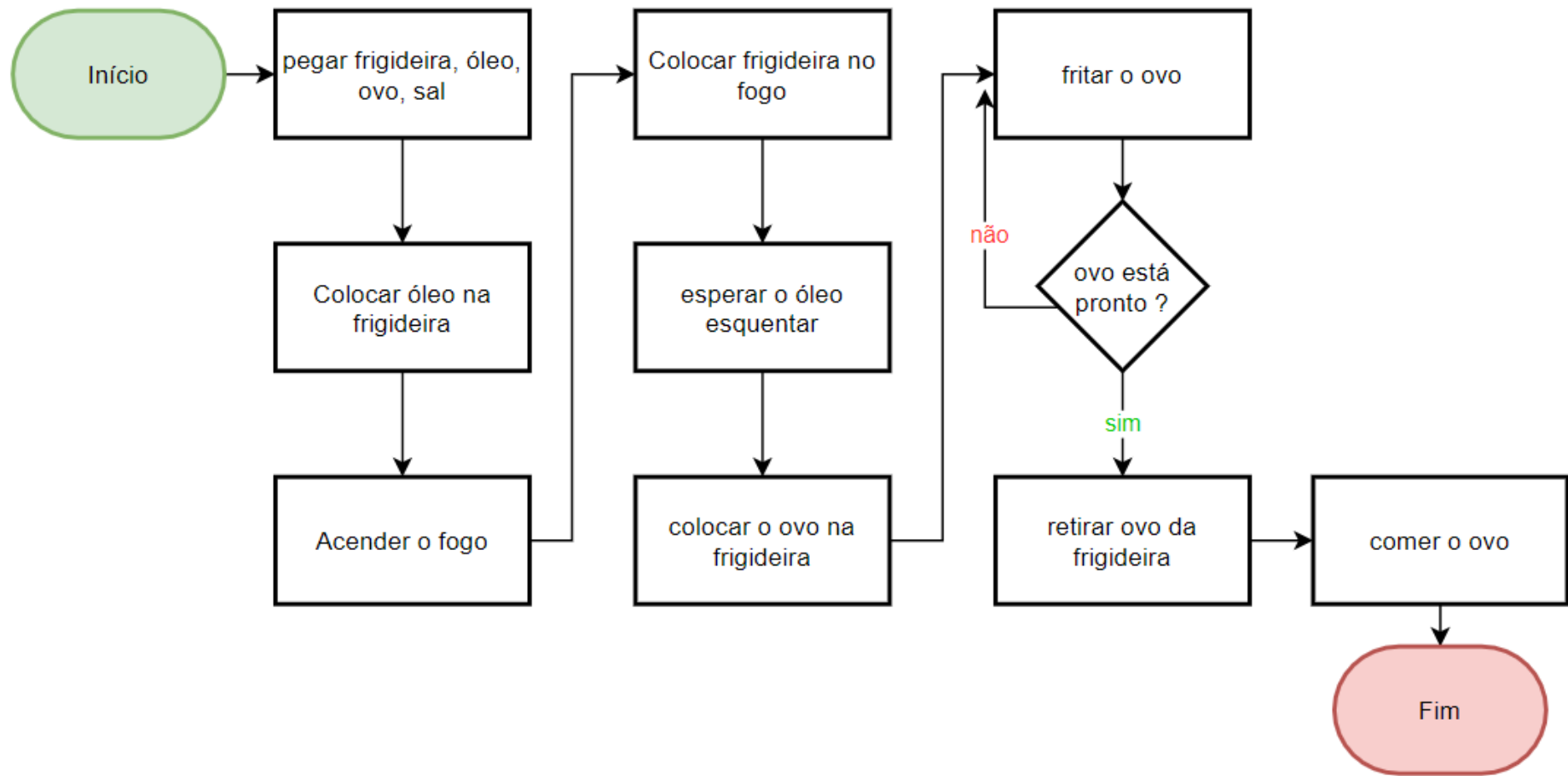
- Utilizando seu caderno, ou o bloco de notas, faça a descrição de um processo **passo a passo**.
- Utilize quantos passos forem necessários.
- Exemplo de processo : **Fritar um ovo**



Fluxo de Processos Descritivo

- Exemplo de processo descritivo **passo a passo**:
- Processo - **Fritar um ovo**
 1. pegar frigideira, óleo, ovo, sal
 2. colocar óleo na frigideira
 3. acender o fogo
 4. colocar a frigideira no fogo
 5. esperar o óleo esquentar
 6. colocar o ovo na frigideira
 7. fritar o ovo
 8. ovo está pronto? **Sim** (vá para linha 9) **Não** (retornar para linha 7)
 9. retirar ovo da frigideira
 10. comer

Fluxograma de Processos








Modelagem de Processos

- Modelagem de processos é a atividade de criar um **diagrama** ou um desenho onde este processo é descrito.
- A modelagem serve para ter uma **visão macro** do processo, facilitar o **entendimento** e **corrigir erros** de um fluxo de processos.
- Abra o Google Diagrams ou draw.io para criar o diagrama:
- <https://www.diagrams.net/>
- Utilize sua conta no Google para realizar o cadastro neste aplicativo.



Fluxogramas

1) Simbologia

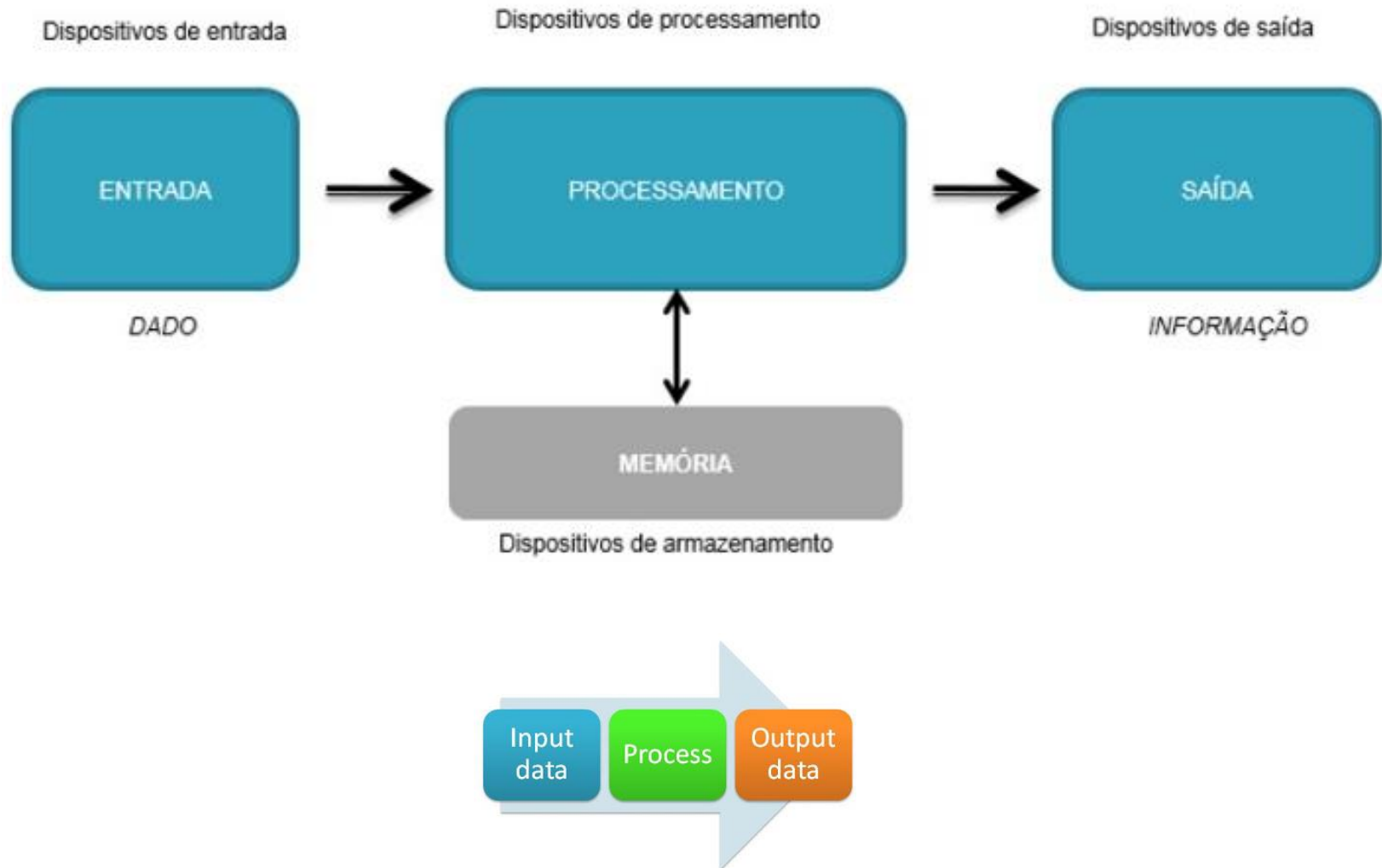
Nomenclatura	Símbolo	Função
Terminal		Indica o INÍCIO ou FIM de um processamento Exemplo: Início do algoritmo
Processamento		Processamento em geral Exemplo: Cálculo de dois números
Entrada de dado manual		Indica entrada de dados através do Teclado Exemplo: Digite a nota da prova 1.
Exibir		Mostra informações ou resultados Exemplo: Mostre o resultado do cálculo
Decisão		Indica quando há alguma decisão a ser tomada, tendo como resposta, somente dois estados: sim ou não.

Modelagem de Processos

- Utilizando o **Google Diagrams**, elabore um diagrama dos processos abaixo (utilize modelo **Flowcharts**):
 - O processo deve ter ser eficaz, eficiente, lógico e completo.
 - Se necessário, faça o descritivo e depois a modelagem (diagrama)
 - Salvar os exercícios utilizando o seguinte padrão: **exe01**
-
- ❖ **Exercício 1 – Fazer um bolo.**
 - ❖ **Exercício 2 – Pegar um ônibus.**
 - ❖ **Exercício 3 – Trocar uma lâmpada.**
 - ❖ **Exercício 4 – Beber água.**
 - ❖ **Exercício 5 – Enviar um e-mail.**

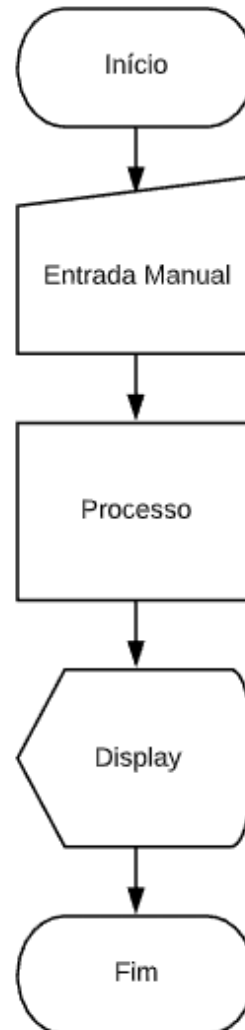


Fluxo de dados no computador



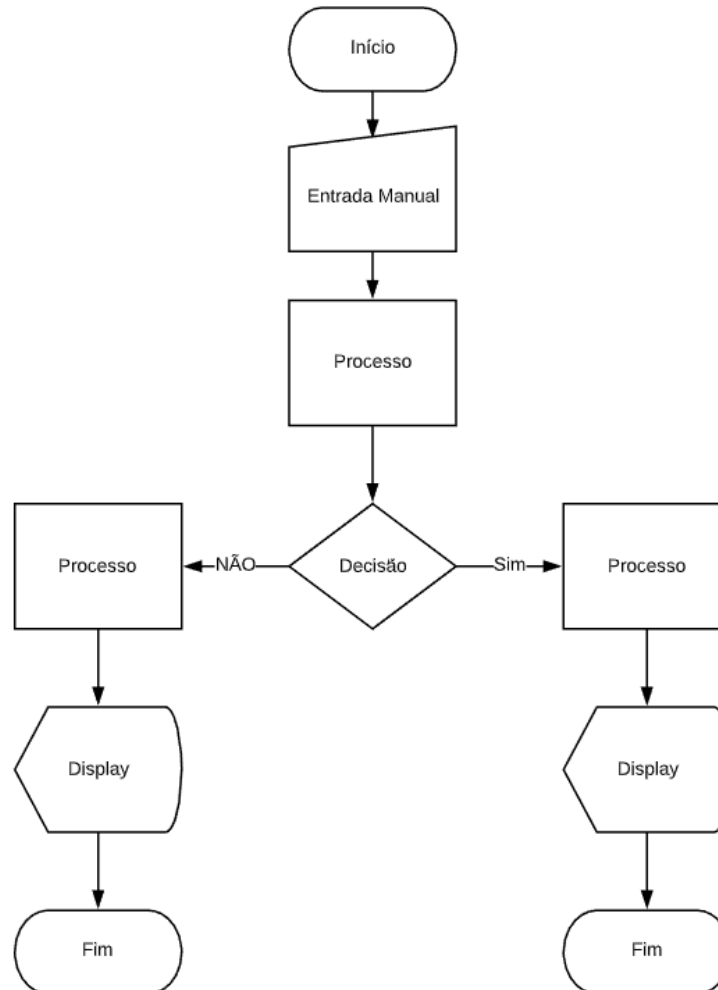
Tipos Básicos de Diagramas

- **Diagrama sequencial**
- **Exemplo 1:**



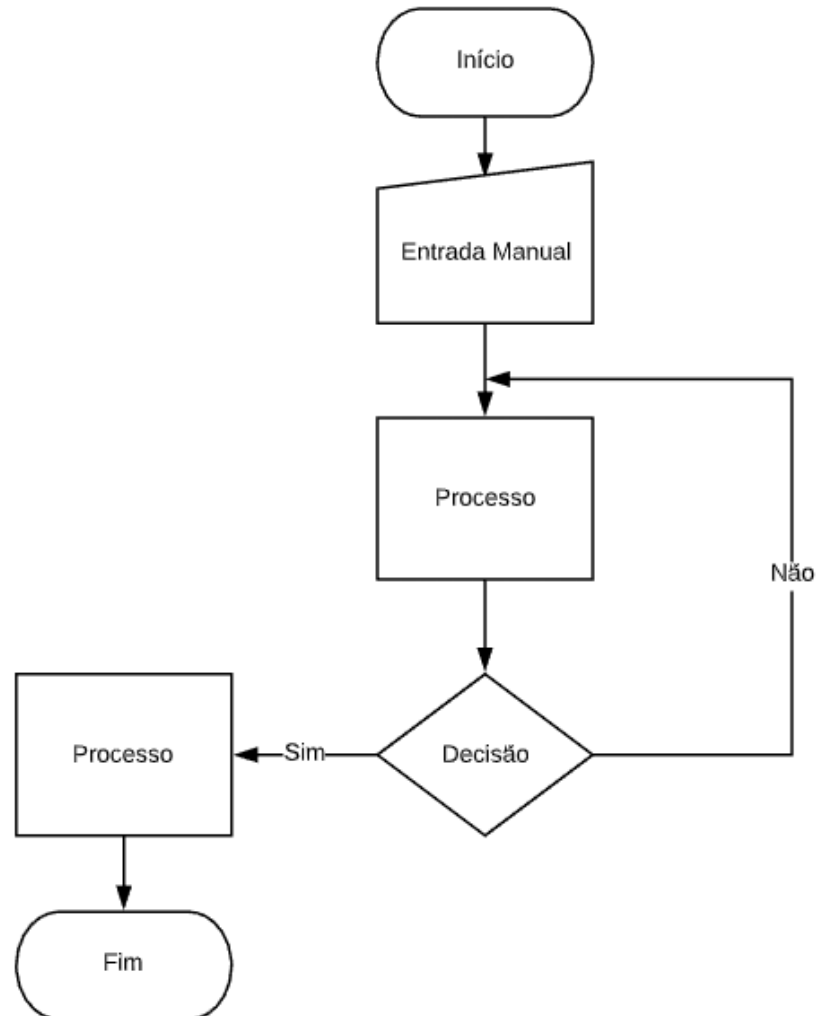
Tipos Básicos de Diagramas

- **Diagrama condicional - decisão**
- **Exemplo 2:**



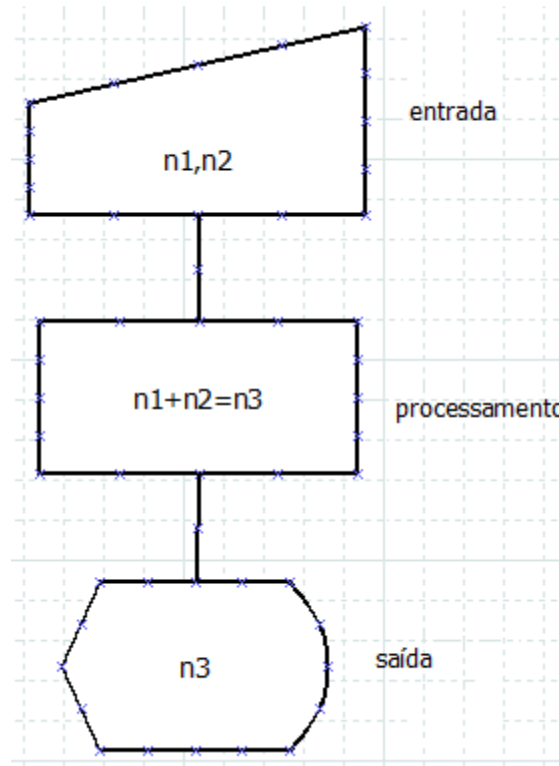
Tipos Básicos de Diagramas

- **Diagrama de uma repetição (loop)**
- **Exemplo 3:**



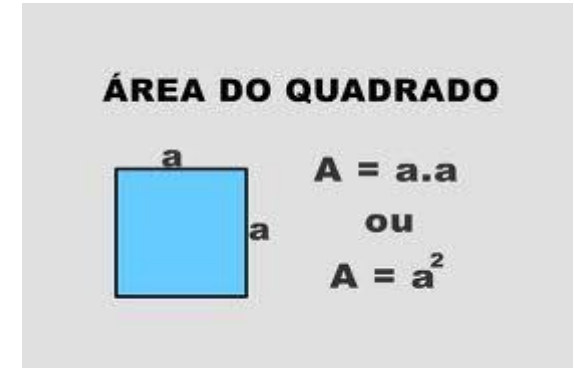
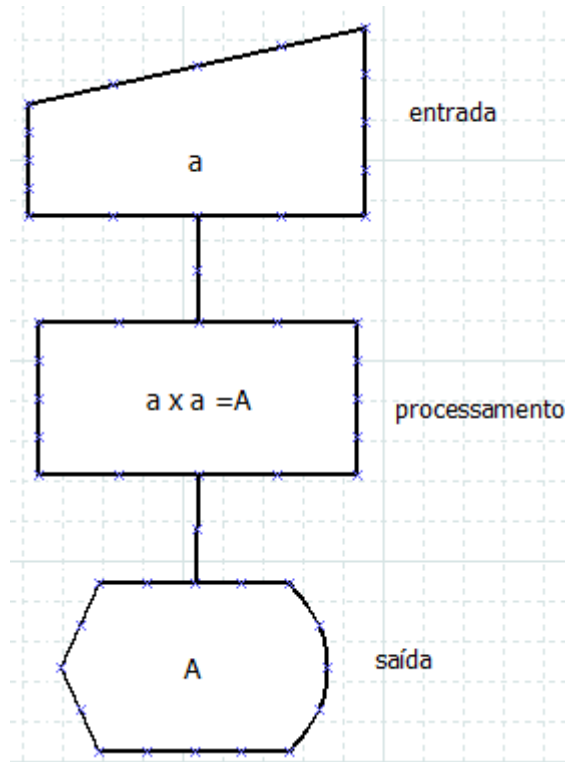
Fluxograma de Dados

- **Exemplo 4:**
- Algoritmo de soma de 2 números:
- **$n1 + n2 = n3$**
- Fluxograma de soma



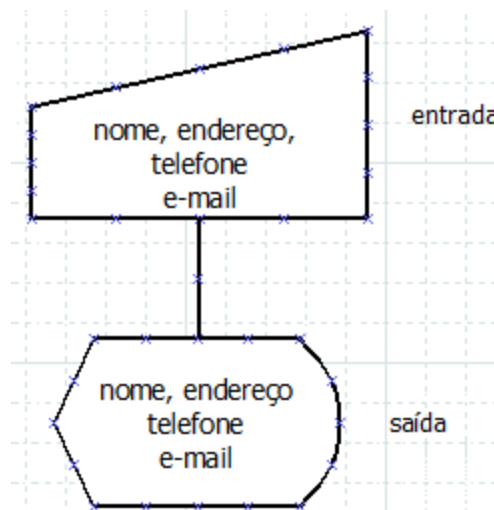
Fluxograma de Dados

- **Exemplo 5:**
- Algoritmo do cálculo da área de um quadrado:
- $a \times a = \text{área}$



Fluxograma de Dados

- **Exemplo 6:**
- Algoritmo que leia as seguintes informações e mostre na tela:
- **Nome, endereço, telefone, e-mail**

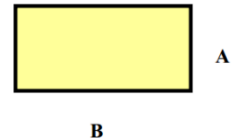


Exercícios

- Utilize o programa **Diagrams** para elaborar os diagramas:

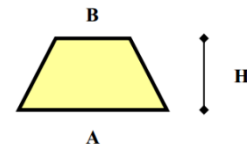
❖ Exercício 6 – Área do retângulo.

- Fazer um fluxograma que solicite o valor dos lados **a** e **b** de um retângulo e calcule a sua área.



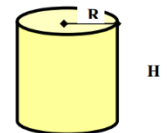
❖ Exercício 7 – Área do trapézio

- Fazer um fluxograma que solicite o valor das bases **a**, **b** e da altura **h** de um trapézio e calcule a sua área.



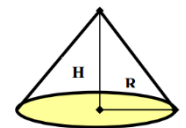
❖ Exercício 8 – Volume do cilindro

- Fazer um fluxograma que solicite o valor do raio **r** e a altura **h** de um cilindro e calcule o seu volume.



❖ Exercício 9 – Volume do cone

- Fazer um fluxograma que solicite o valor do raio **r** e a altura **h** de um cone e calcule o seu volume.



Exercícios

- Utilize o programa **Diagrams** para elaborar os diagramas abaixo:

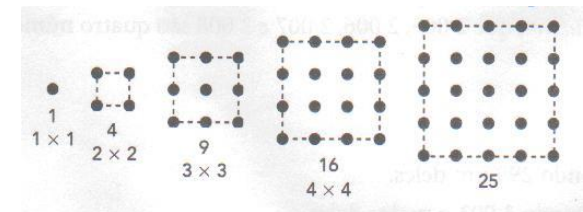
❖ Exercício 10 – Conversão de dólar para real

1. Informar o valor da cotação do dólar atual.
2. Informar a quantidade de dólares que deseja converter
3. Converter para reais e mostrar a informação.
4. Apresentar o resultado da conversão



❖ Exercício 11 – Soma dos quadrados dos números

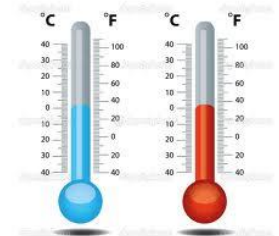
1. Informar o valor de 4 números quaisquer.
2. Elevar ao quadrado cada um destes números.
3. Somar o valor dos quadrados dos 4 números.
4. Apresentar o resultado final da soma dos quadrados.



- Utilize o programa **Diagrams** para elaborar os diagramas abaixo:

❖ Exercício 12 – Conversão de temperatura

- Informar uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit.
- A fórmula de conversão é: $F = (9 * C / 5) + 32$, sendo **F** a temperatura em Fahrenheit e **C** a temperatura em Celsius.



❖ Exercício 13 – Salário líquido

- Faça um algoritmo que solicite o nome e o **salário bruto (sem impostos)** de um funcionário de uma empresa.
- Sabe-se que o imposto de renda é de 27% sobre o salário.
- Calcule e mostre o **salário líquido (sem impostos)**
- Ao final informe o nome e o salário líquido do funcionário.



- Utilize o programa **Diagrams** para elaborar os diagramas abaixo:
- ❖ **Exercício 14 – Comissão de vendedoras de roupas**
 1. Construa um fluxograma para calcular o valor da comissão de uma vendedora de roupas.
 2. A sua comissão será de **12%** do total de vendas realizado
 3. Sabe-se que a loja vende três tipos de roupa: **calça, blusa e saia**.
- Você deve informar os seguintes dados (variáveis):
 - a) **Nome** da vendedora
 - b) **Preço unitário** de cada roupa (calça, blusa, saia)
 - c) **Quantidade** vendida de cada roupa (calça, blusa, saia)
- Ao final mostre o **nome da vendedora** e a **comissão da vendedora**.



Exercícios

❖ Exercício 15 – Custo da pintura de uma sala

- Construa um fluxograma para calcular o **custo da pintura** de uma sala.
- Considere que a sala não possui portas ou janelas.
- Para efetuar este cálculo você precisa coletar os seguintes dados:
 1. **Medidas da sala:** largura, comprimento e altura das paredes e do teto.
 2. **Quantidade de demão utilizada na pintura.**
 3. **Rendimento de uma lata de tinta por metro quadrado.** (pesquise !)
 4. **Preço de uma lata de tinta.** (pesquise !)
 5. **Custo da mão de obra do pintor por m².**
- Calcule e mostre:
 1. **Quantidade de latas de tinta necessária.**
 2. **O custo com a tinta e custo da mão de obra do pintor.**
 3. **O custo total da obra (pintura) a ser executada.**

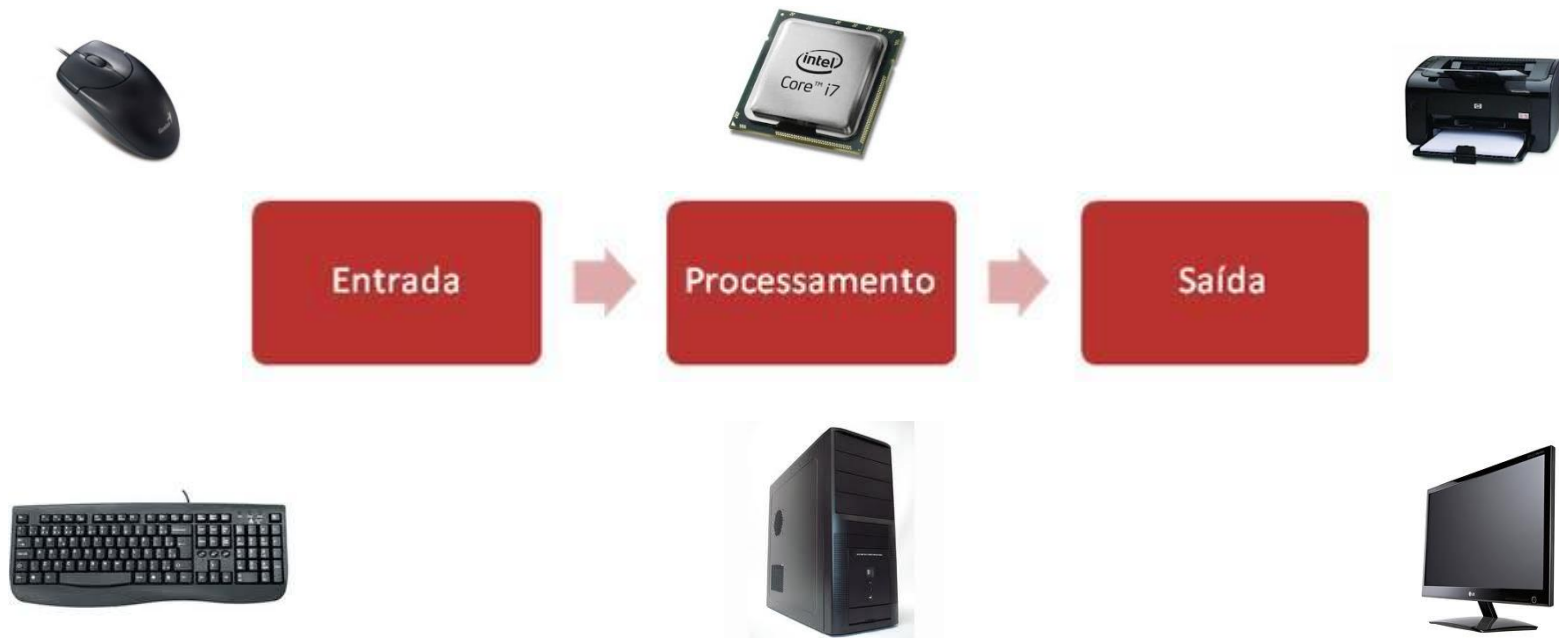


Aula 3 – Algoritmos e Programação Básica

```
HELLO, WORLD!  
  
LIST  
10 HOME  
20 INVERSE  
30 PRINT "HELLO, WORLD!"  
40 NORMAL  
50 PRINT CHR# (7)  
■
```

Algoritmos e Programação

- Um **algoritmo** é uma sequência finita de instruções bem definidas.
- Um programa de computador é essencialmente um algoritmo que diz ao computador os passos específicos e em que ordem eles devem ser executados.
- Um programa de computador possui, basicamente, **entrada** de dados do algoritmo, **processamento** de dados e **saída** de dados.

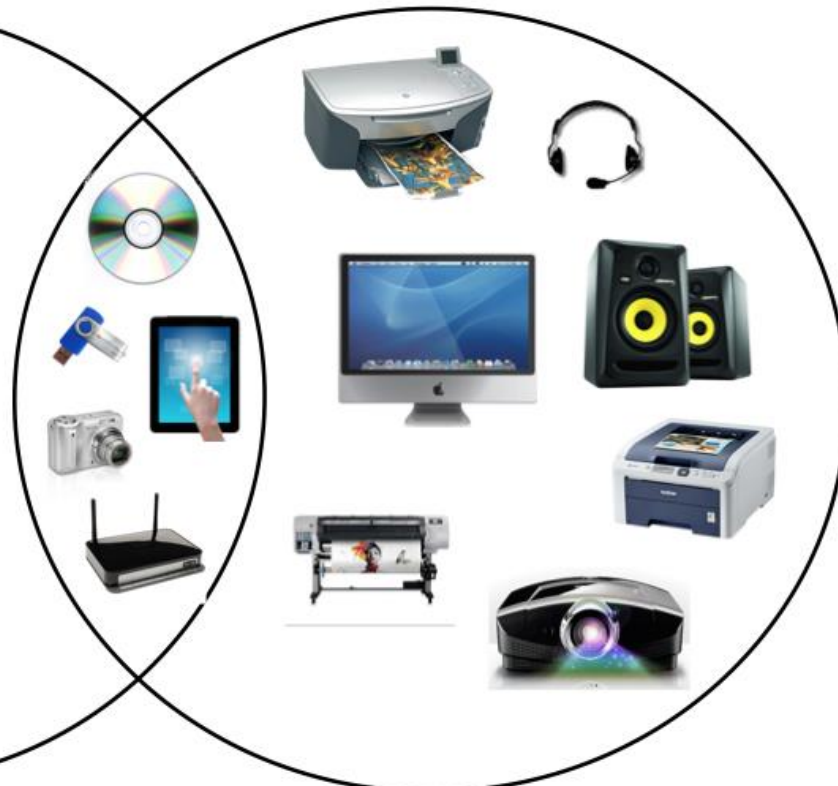


Dispositivos de Entrada e Saída

Input



Output



Pseudocódigos

- **Pseudocódigo**
- Pseudocódigo é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples (nativa a quem o escreve, de forma a ser entendida por qualquer pessoa) sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação.
- É, como o nome indica, um pseudocódigo e, portanto, não pode ser executado num sistema real (computador) — de outra forma deixaria de ser pseudo.
- Os livros sobre a ciência de computação utilizam frequentemente o pseudocódigo para ilustrar os seus exemplos, de forma que todos os programadores possam entendê-los (independentemente da linguagem que utilizem).
- **Portugol**
- Portugol, também conhecido como **Português Estruturado**, é um pseudocódigo escrito em português.

Portugol Studio

- **Portugol Studio**
- O Portugol Studio é uma ferramenta para **aprender programação**, voltada para as pessoas que falam o idioma português. Possui uma sintaxe fácil baseada em C e PHP, diversos exemplos e materiais de apoio à aprendizagem. Também possibilita a criação de jogos e outras aplicações.
- Foram feitos alguns estudos de como o aprendizado de programação pode ser difícil para iniciantes, principalmente usando IDEs profissionais. Além disso, as linguagens de programação usam, na maioria, o idioma inglês, uma barreira muito comum para iniciantes. Além de que as mensagens de erro de muitos compiladores não indicam o problema dificultando saber qual erro o usuário iniciante está cometendo. Outro problema é o fato de os depuradores das IDEs profissionais serem muito complexos, dificultando ao aprendiz saber o que está acontecendo.
- Link de acesso ao Portugol Studio:
- <http://lite.acad.univali.br/portugol/>

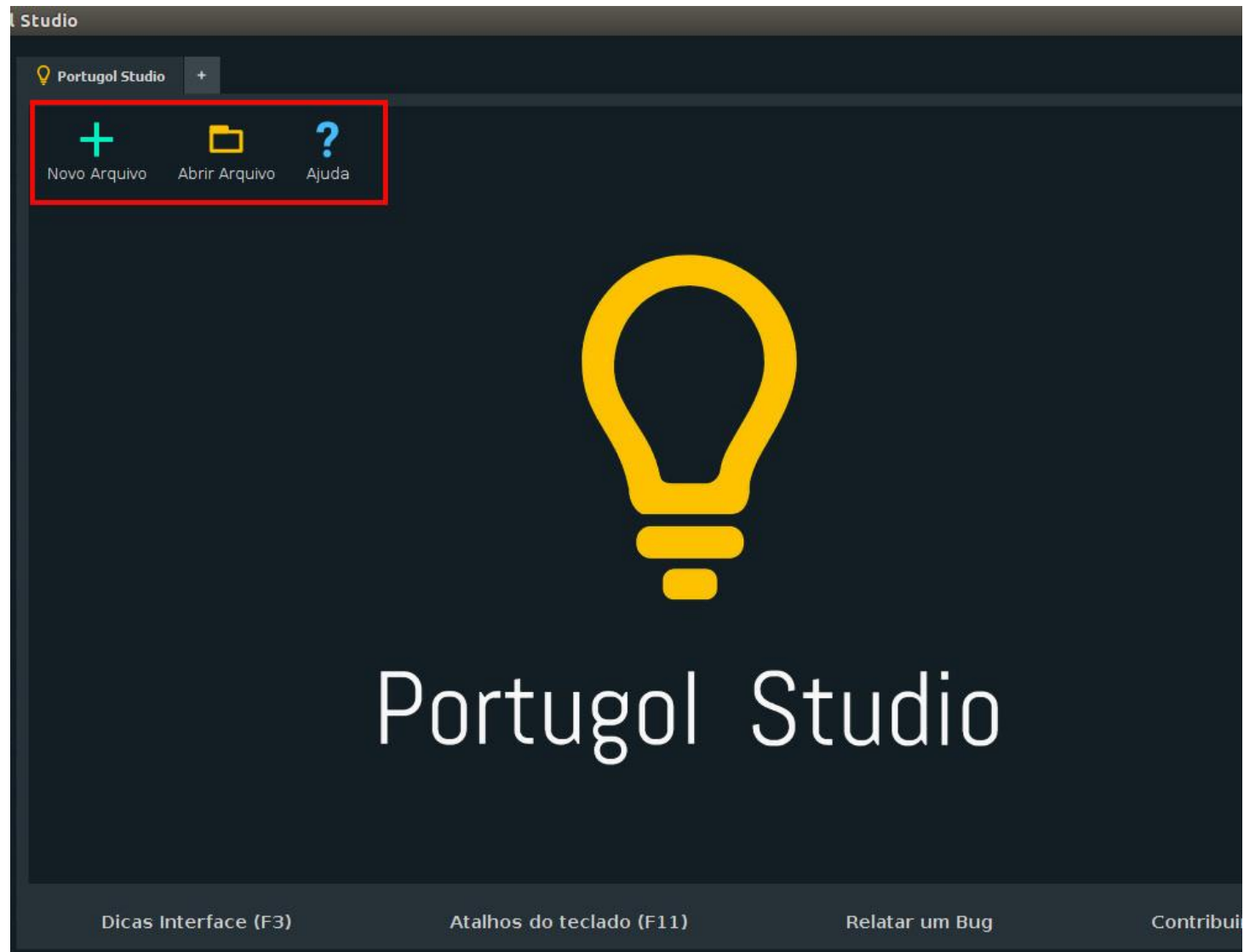
{PortugolStudio}

Linguagem Portugol

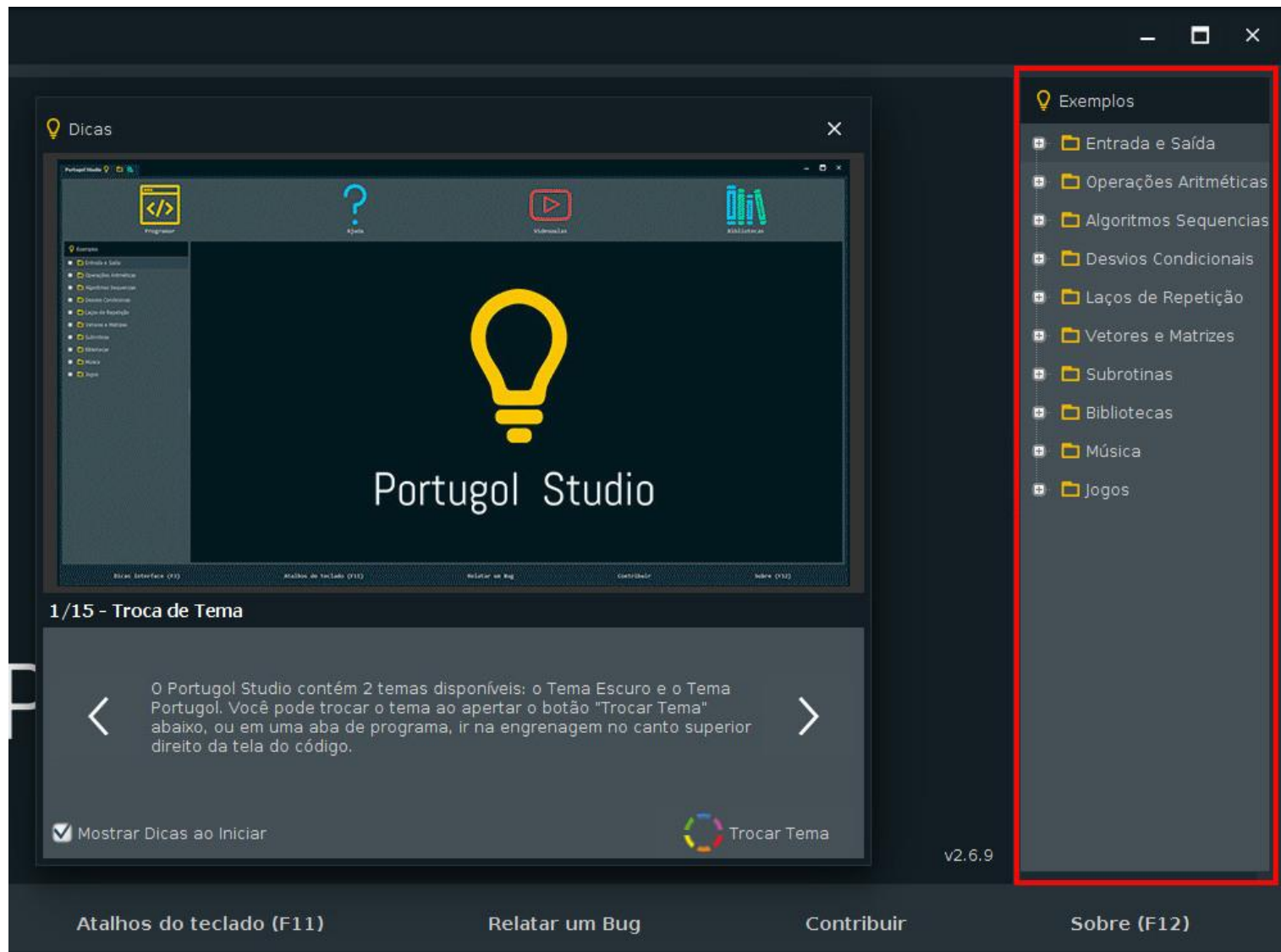
- **Portugol Webstudio**
- Uma linguagem de programação é um método padronizado para comunicar instruções para um computador.
- É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.
- O Portugol é uma representação que se assemelha bastante com a linguagem C, porém é escrito em português. A ideia é facilitar a construção e a leitura dos algoritmos usando uma linguagem mais fácil aos alunos.
- Link de acesso ao Portugol Webstudio:
- <https://portugol-webstudio.cubos.io/ide>



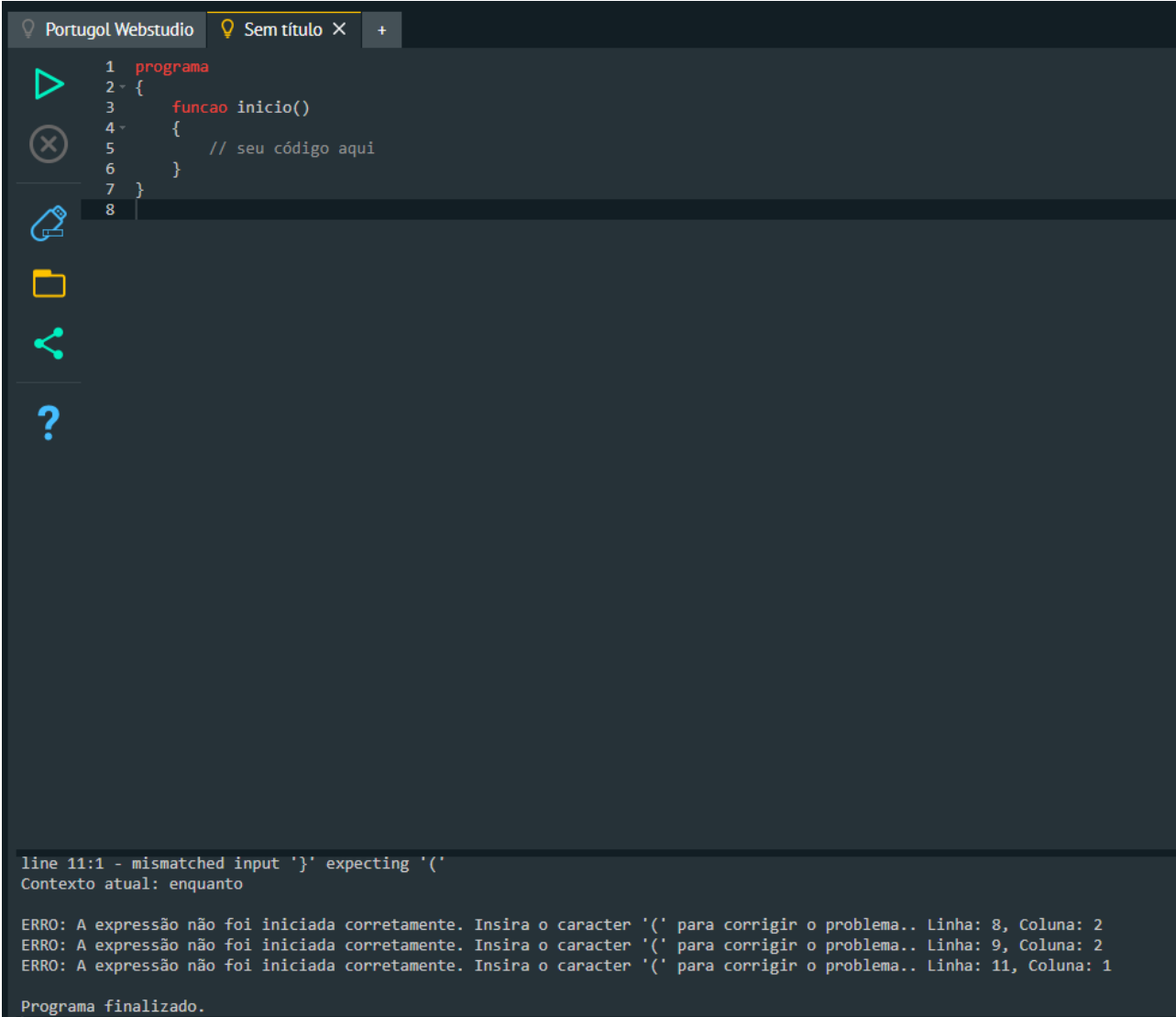
Conhecendo o Portugol Studio



Conhecendo o Portugal Studio



Conhecendo o Portugol Studio



```
1 programa
2 {
3     funcao inicio()
4     {
5         // seu código aqui
6     }
7 }
8
```

line 11:1 - mismatched input '}' expecting '('
Contexto atual: enquanto

ERRO: A expressão não foi iniciada corretamente. Insira o caracter '(' para corrigir o problema.. Linha: 8, Coluna: 2
ERRO: A expressão não foi iniciada corretamente. Insira o caracter '(' para corrigir o problema.. Linha: 9, Coluna: 2
ERRO: A expressão não foi iniciada corretamente. Insira o caracter '(' para corrigir o problema.. Linha: 11, Coluna: 1

Programa finalizado.

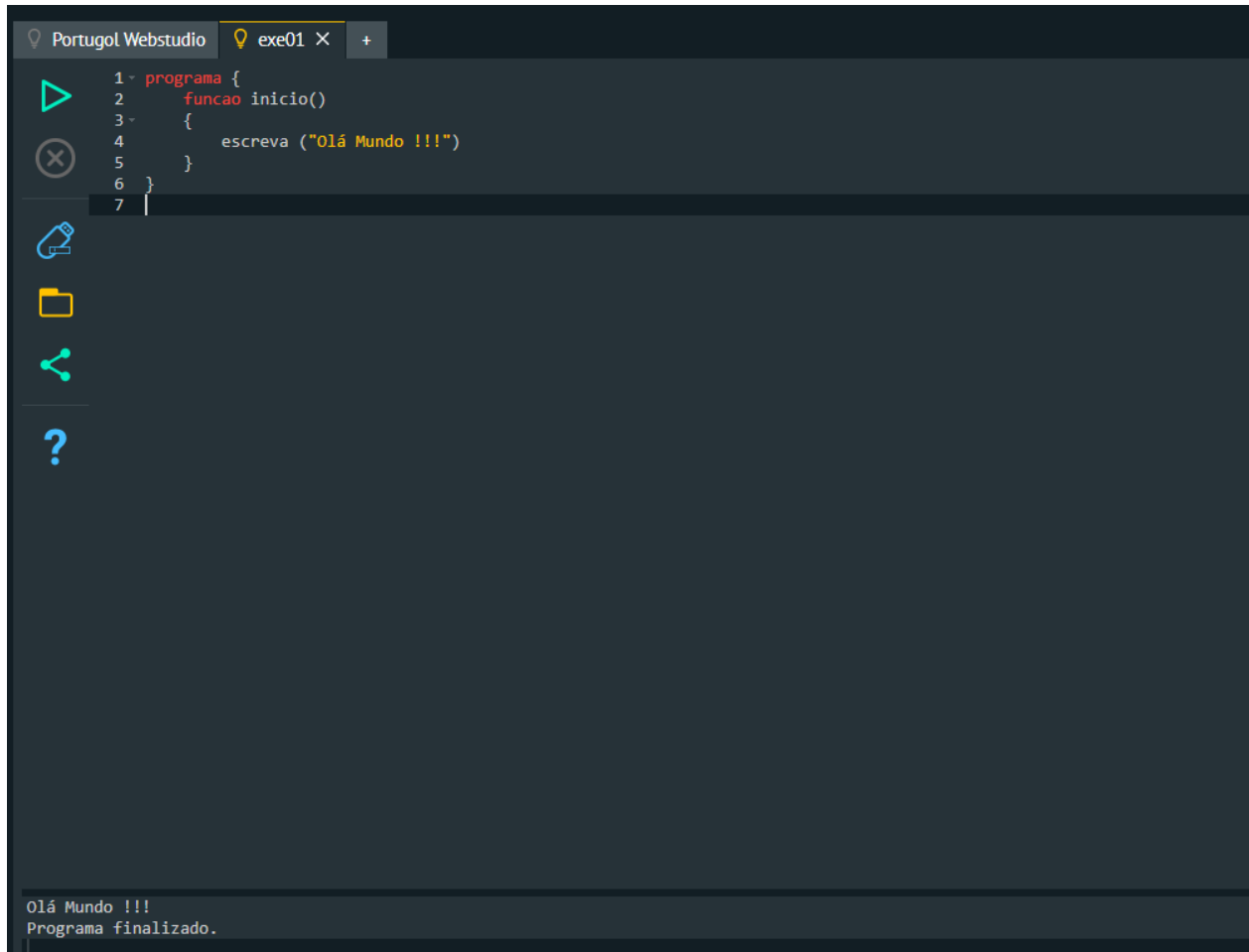
Hello World !

- Chegou a hora de criar o seu **primeiro programa** no Portugol Studio.
- Primeiramente, abra o Portugol.
- Crie um novo arquivo.
- Como você verá, ele já tem uma **estrutura inicial**.
- Na estrutura inicial é marcado o início e o fim do programa com chaves **{ }**
- A estrutura inicial também marca o início e o fim da função principal do programa com chaves **{ }**
- Copie o exemplo do "Hello, World!" em português:

```
1  programa
2  {
3      funcao inicio()
4      {
5          escreva("Olá mundo !")
6      }
7  }
```

Hello World !

- Faça o algoritmo do “Hello World” em português no Portugol Studio:
- Depois clique no ícone **play** para executar o programa:



The screenshot shows the Portugol Webstudio IDE interface. At the top, there's a tab labeled 'Portugol Webstudio' and another labeled 'exe01' with a close button. The main editor area contains the following code:

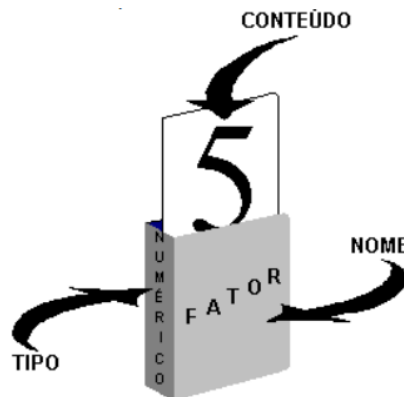
```
1- programa {  
2-   funcao inicio()  
3-   {  
4-     escreva ("Olá Mundo !!!")  
5-   }  
6- }  
7- |
```

On the left side of the editor, there's a vertical toolbar with icons for running (a green play button), closing (a red X), debugging (a blue bug), file explorer (a yellow folder), sharing (a green share icon), and help (a blue question mark). At the bottom of the IDE, there's a console window showing the output of the program:

```
Olá Mundo !!!  
Programa finalizado.
```

Variáveis

- Uma **variável** é uma **informação que fica armazenada na memória**. Pode ser um número, um texto, dentre outras coisas.
- Uma variável pode ser vista como uma caixa com um rótulo ou nome colado a ela, que num dado instante **guarda um determinado** objeto. O conteúdo desta caixa não é algo fixo, permanente. Na verdade, **essa caixa pode ter seu conteúdo alterado diversas vezes**. Contudo, o conteúdo deve ser **sempre do mesmo tipo**.
- Na figura abaixo, a caixa (variável) rotulada com FATOR contém o valor 5. Como seu tipo é **numérico**, em um determinado instante essa caixa poderá conter qualquer valor numérico (inteiro ou fracionário; positivo, negativo ou zero). Entretanto, em um determinado instante, ela conterà um, e somente um, valor.



Declaração e Atribuição de Variáveis

- Em muitas linguagens de programação, antes que você possa usar qualquer variável, você deve **declará-la**.
- Uma declaração é uma instrução que fornece um **identificador** para uma **variável**, precedido pelo **tipo da variável**. É o tipo que determina o **conjunto de valores que a variável pode assumir**. O identificador é simplesmente o nome da variável.
- Exemplo de declarações em Portugol Studio:

```
1  programa
2  {
3      funcao inicio()
4      {
5          // tipo e identificador da variável:
6          caracter nome_variavel
7          inteiro variavel_inicializada = 42
8          real nome_variavel2
9          logico nome_variavel3
10         // ou para declarar varias variáveis de um mesmo tipo:
11         cadeia var1,var2,var3,var4
12         logico var4,var5,var6
13     }
14 }
```


Tipos de Dados

- Quais são os **tipos de dados** que o computador pode armazenar?
- Sendo assim, existem alguns tipos básicos de dados nos quais valores podem ser armazenados no computador. O Portugol exige que o tipo de dado de um valor seja do mesmo tipo da variável ao qual este valor será atribuído.
- **Tipo Inteiro**
- Uma variável do tipo inteiro pode ser entendida como uma variável que contém **qualquer número que pertença ao conjunto dos números inteiros**. Podem ser positivos, negativos ou nulos.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro num1, num2
6          num1 = 5
7          num2 = 3
8          escreva (num1 + num2)
9      }
10 }
```

Tipos de Dados

- **Tipo Real**
- Em algumas situações é necessário armazenar valores que não pertencem aos números inteiros. Por exemplo, se quiséssemos armazenar o valor da divisão de 8 por 3 ?
- Uma variável do **tipo real armazena um número real como uma fração decimal** possivelmente infinita, como o número PI 3.1415926535.
- Os valores do tipo de dado real são números separados **por pontos** e não por virgulas.

```
1  programa
2  {
3      funcao inicio()
4      {
5          real div
6
7          div = 8.0/3.0
8
9          escreva (div)
10     }
11 }
```

Tipos de Dados

- **Tipo Character**
- Em determinadas situações faz-se necessário o uso de símbolos, letras ou outro tipo de conteúdo. Por exemplo, em um jogo da velha, seriam necessárias variáveis que tivessem conteúdos de 'X' e 'O'.
- Para este tipo de situação, existe a variável do tipo **character**. A variável do tipo character é aquela que **contém uma informação composta de apenas UM carácter** alfanumérico ou especial.
- Exemplos de **caracteres são letras, números, pontuações** e etc.

```
1  programa
2  {
3      funcao inicio()
4      {
5          caracter vogal, consoante
6          vogal = 'a'          //variável atribuida pelo programador
7
8          escreva ("Digite uma consoante: ")
9          leia (consoante)     //variável declarada através de entrada do usuário
10
11         escreva ("Vogal: ", vogal, "\n", "Consoante: ", consoante)
12     }
13 }
```

Tipos de Dados

- **Tipo Cadeia**
- Em algumas situações precisa-se armazenar em uma variável, um texto ou uma **quantidade grande de caracteres**.
- Para armazenar este tipo de conteúdo, utiliza-se uma variável do tipo cadeia.
- Cadeia é uma **sequência ordenada de caracteres** (símbolos) escolhidos a partir de um conjunto pré-determinado.

```
1  programa
2  {
3      funcao inicio()
4      {
5          cadeia nome1, nome2
6
7          nome1 = "Variável declarada através de atribuição" //atribuído pelo programador
8
9          escreva ("Digite seu nome: ")
10         leia (nome2) //variável declarada através de entrada do usuário
11         escreva ("\nOlá ", nome2)
12     }
13 }
```

Saída de Dados: escreva()

- No Portugol Studio, o comando **escreva()** é usado para imprimir alguma coisa na tela. Para mostrar um texto na tela use o seguinte padrão:

```
1 programa
2 {
3     funcao inicio()
4     {
5         escreva("Digite aqui seu texto !")
6     }
7 }
```

- Porém, você viu apenas um uso básico do comando, onde foi usado um texto comum. Agora, você vai se aprofundar mais no comando.
- Primeiramente, você pode usar `\n` para pular uma linha e `\t` para dar um tab no texto. Faça o exemplo abaixo:

```
1 programa
2 {
3     funcao inicio()
4     {
5         escreva("Primeira linha\nSegunda linha\n")
6     }
7 }
```

Saída de Dados: escreva()

- Uma função muito importante é escrever ou mostrar o valor contido em uma variável utilizando o comando **escreva()**.
- No código abaixo, a variável pontos foi declarada e inicializada com o valor 10. Depois, usamos o comando **escreva()** para imprimi-la na tela. Note que a palavra pontos aparece duas vezes nesse comando.
- Na primeira vez, ele faz referência à variável pontos que foi declarada. Na segunda, ela imprime a palavra pontos literalmente.
- Se estiver no meio do texto, a variável deve ser precedida e antecedida de vírgula. Se a variável estiver no início deve ser precedida de vírgula e se estiver no fim do texto deve ser antecedido de vírgula.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro pontos = 10
6          escreva("Você fez ", pontos, " pontos!\n")
7      }
8  }
```

Entrada de Dados: leia()

- A entrada de dados é feita com o comando **leia()**. Assim como o comando `escreva()`, ele também aceita vários argumentos.
- Perceba que eu declarei algumas variáveis (três inteiros), mas não atribuí nenhum valor a elas. Eu permiti que o usuário definisse o valor delas pelo comando **leia()**.
- Esse comando só aceita variáveis como argumentos. O que o usuário digitar ele vai armazenar nas variáveis.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro x, y, idade
6          escreva("Digite a sua idade\n")
7          leia(idade)
8          escreva("Digite as suas coordenadas\n")
9          leia(x, y)
10         escreva("Coordenadas:\n", x, " e ", y, "\n", "Idade:\n", idade, "\n")
11     }
12 }
```


Limpando a Tela

- Para limpar a tela do console, use o comando **limpa()**.
- Faça um teste: coloque esse comando antes do comando que mostra as variáveis do exemplo da idade e das coordenadas da seção anterior.
- Este comando é utilizado para separar os dados de entrada (dados que são solicitados ao usuário do programa) dos dados de saída ou os resultados do programa.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro x, y, idade
6          //dados de entrada
7          escreva("Digite a sua idade\n")
8          leia(idade)
9          escreva("Digite as suas coordenadas\n")
10         leia(x, y)
11         limpa()
12         //dados de saída
13         escreva("Coordenadas:\n", x, " e ", y, "\n", "Idade:\n", idade, "\n")
14     }
15 }
```

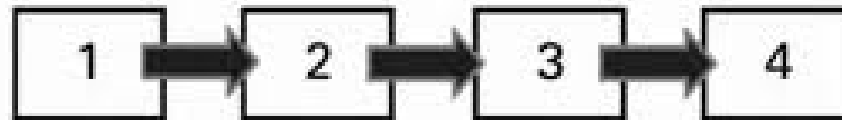
Aula 4 – Algoritmos Sequenciais

```
HELLO, WORLD!  
  
LIST  
10 HOME  
20 INVERSE  
30 PRINT "HELLO, WORLD!"  
40 NORMAL  
50 PRINT CHR# (7)  
■
```

Algoritmos Sequenciais

- **Algoritmos Sequenciais**

1. A execução das tarefas é corretamente cumprida, se executarmos todos os passos (instruções) na sequência em que elas aparecem, da primeira até a última, sem omissões e sem repetições.
2. De forma genérica, a construção de um algoritmo se resume às seguintes etapas:
 - a) Entendimento do problema;
 - b) Elaboração da solução algorítmica; e
 - c) Codificação da solução no Português Estruturado;



Operadores de Atribuição

O operador de atribuição é composto apenas pelo **=**

O operador da atribuição sempre opera da direita para a esquerda.

Isso quer dizer que o valor da expressão à direita do operador de atribuição é avaliado primeiro, então o resultado é atribuído ao operando na esquerda.

x = 7

x = 4 * 5

x = y + 8

x = y

Veja abaixo alguns exemplos de atribuições inválidas:

7 * 3 = x

x - 8 = 10

x / 4 = y

Operadores Aritméticos

Os operadores aritméticos são: **+** **-** ***** **/** **%**

Estes cinco operadores representam as quatro operações básicas da matemática e a operação de módulo, que retorna o resto da divisão do primeiro operando pelo segundo.

Veja exemplos de cada uma dessas operações:

Adição: $x = 5 + 6$

Subtração: $x = 7 - 4$

Multiplicação: $x = 5 * 3$

Divisão: $x = 10 / 5$

Módulo*: $x = 13 \% 5$

* Módulo é o resto da divisão por inteiro de um número

Algoritmos Sequenciais

- **Algoritmos sequenciais utilizando tipos primitivos de dados.**
- Os tipos de dados básicos ou primitivos, se encaixam em duas categorias: **numérica** (que engloba números inteiros ou reais) e **caractere** (que guarda um caractere, que pode ser uma letra, um número, um símbolo, etc...).
- **Exemplo 1 – Tipo caractere (um caractere)**
 1. Elabore um programa que exiba na tela o caractere especial “@”
 2. Não é necessário o uso de variáveis para este programa

```
1 programa
2 {
3     funcao inicio()
4     {
5         escreva ("&")
6     }
7 }
```

Algoritmos Sequenciais

- **Exemplo 2 – Tipo caractere (palavra ou frase)**
 1. Elabore um programa que exiba na tela as frases abaixo:
 2. **“Eu sou um programador”**
 3. Pular linha inserindo **\n** no final do texto e exibir outra frase:
 4. **“Instituto Federal de São Paulo”**
 5. A frase acima utiliza caracteres especiais da língua portuguesa.
 6. Não é necessário o uso de variáveis para este programa

```
1  programa
2  {
3      funcao inicio()
4      {
5          escreva ("Eu sou um programador\n")
6          escreva ("Instituto Federal de São Paulo")
7      }
8  }
```


Algoritmos Sequenciais

- **Exemplo 3 – Tipo cadeia de caracteres (palavra ou frase)**
 1. Elabore um programa que solicite o **nome do usuário**
 2. Limpe a tela e mostre este dado no final.
 3. Este programa necessita declaração de variáveis.
 4. A variável **nome** é do tipo **cadeia de caracteres**.

```
1  programa
2  {
3      funcao inicio()
4      {
5          cadeia nome
6          escreva ("Por favor, digite seu nome:\n")
7          leia (nome)
8          limpa ()
9          escreva ("Meu nome é ",nome)
10     }
11 }
```

Algoritmos Sequenciais

- Exemplo 4 – Tipo numérico inteiro

1. Elabore um programa que solicite o **nome do usuário** e **idade** de uma pessoa qualquer e mostre estes mesmos dados no final.
2. A variável **nome** é do tipo **cadeia**.
3. A variável **idade** é do tipo **inteiro**.

```
1  programa
2  {
3      funcao inicio()
4      {
5          cadeia nome
6          inteiro idade
7          escreva ("Por favor, digite seu nome:\n")
8          leia (nome)
9          escreva (nome, ", por favor digite a sua idade:\n")
10         leia (idade)
11         limpa()
12         escreva ("Meu nome é ", nome, " e minha idade é ", idade)
13     }
14 }
```

Algoritmos Sequenciais

- **Exemplo 5 – Tipo numérico real**
1. Elabore um programa que solicite o **nome do usuário**, **idade** e **altura** de uma pessoa qualquer e mostre estes mesmos dados no final.
 2. A variável **nome** é do tipo **cadeia**.
 3. A variável **idade** é do tipo **inteiro** e a variável **altura** é do tipo **real**.

```
1  programa
2  {
3      funcao inicio()
4      {
5          cadeia nome
6          inteiro idade
7          real altura
8          escreva ("Por favor, digite seu nome:\n")
9          leia (nome)
10         escreva (nome,", por favor digite a sua idade:\n")
11         leia (idade)
12         escreva (nome,", por favor digite a sua altura:\n")
13         leia (altura)
14         limpa ()
15         escreva ("Meu nome é ",nome,"\t")
16         escreva ("minha idade é ",idade," anos\t")
17         escreva ("e minha altura é ",altura," metros")
18     }
19 }
```

Algoritmos Sequenciais

- Exemplo 6 – Executando expressões matemáticas e cálculos
- Elabore um programa que solicite 2 números e realize as 4 operações básicas da matemática: **soma**, **subtração**, **multiplicação** e **divisão**.

```
1  programa
2  {
3      funcao inicio()
4      {
5          real n1,n2,soma,subt,mult,divs
6          escreva ("Digite o valor do 1º número:\n")
7          leia (n1)
8          escreva ("Digite o valor do 2º número:\n")
9          leia (n2)
10         soma = n1+n2
11         subt = n1-n2
12         mult = n1*n2
13         divs = n1/n2
14         limpa()
15         escreva ("\nA soma de n1 + n2 é ",soma)
16         escreva ("\nA subtração de n1 - n2 é ",subt)
17         escreva ("\nA multiplicação de n1 * n2 é ",mult)
18         escreva ("\nA divisão entre n1 / n2 é ",divs)
19     }
20 }
```

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe16.por**

❖ Exercício 16 – Escrevendo uma frase na tela

1. Crie um programa que escreva “**Automação Industrial do IFSP !!!**” na tela
- **Objetivo:** mostrar na tela a frase.
 - **Comentário:** verifique a utilização do comando **escreva** e como deve ser o procedimento para escrever uma mensagem na tela. Não são necessárias variáveis neste programa. Sempre verifique o resultado.
 - **Funcionamento geral:**
 - 1) Abra o Portugol Studio e crie um novo arquivo
 - 2) Escreva o programa seguindo os padrões da linguagem
 - 3) Execute e teste o funcionamento do programa

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe17.por**

❖ Exercício 17 – Escrevendo uma mensagem na tela

1. Faça um programa que leia o nome de uma pessoa e mostre uma mensagem de boas vindas
- **Objetivo:** Solicitar o nome e mostrar na tela a mensagem.
 - **Comentário:** verifique a utilização do comando **leia** para solicitar e armazenar o nome. Verifique também o funcionamento do comando **escreva**. É necessário uma variável neste programa para armazenar o nome.
 - **Funcionamento geral:**
 - 1) Abra o Portugol Studio e crie um novo arquivo
 - 2) Escreva o programa seguindo os padrões da linguagem
 - 3) Execute e teste o funcionamento do programa

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe18.por**

❖ Exercício 18 – Soma de 3 números inteiros

1. Elabore um algoritmo que leia três valores inteiros.
 2. Mostre a soma destes valores.
- **Objetivo:** realizar a soma de três números inteiros e mostrar o resultado para o usuário.
 - **Comentário:** como existem três números, cada um deles deve estar armazenado em um local da memória, então devem ser criadas três variáveis para armazenar cada um deles. Além disso, deve-se criar uma variável para armazenar o resultado.
 - **Funcionamento geral:** obtêm-se os três números do usuário; realiza-se a soma; mostra o resultado para o usuário.

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe19.por**

❖ Exercício 19 – Mostrar a multiplicação de dois números inteiros

1. Elabore um algoritmo que leia dois números inteiros e mostre o resultado da multiplicação destes números. Por exemplo, se o usuário digitar o número 2 e 8, o programa deverá mostrar como resultado o valor 16.
- **Objetivo:** calcular a multiplicação (número x número ou use $n*n$) de dois números informados pelo usuário e apresentar o resultado.
 - **Comentário:** o usuário informará dois números, então é necessário duas variáveis para armazenar estes números. Além disso, será feito um cálculo com este número, então é preciso de outra variável para armazenar o resultado.
 - **Funcionamento geral:** solicita-se dois números para o usuário; calcula-se a multiplicação destes; mostra-se o resultado para o usuário.

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe20.por**

❖ Exercício 20 – Multiplicação de 3 números reais dividido por 2

1. Elabore um algoritmo que leia 3 números reais.
 2. Mostre a multiplicação destes dividido por 2.
- **Objetivo:** realizar a multiplicação de três números reais, dividir por dois e apresentar o resultado.
 - **Comentário:** como existem três números, cada um deles deve estar armazenado em um local da memória, então devem ser criadas três variáveis para armazenar cada um deles. Além disso, deve-se criar uma variável para armazenar o resultado.
 - **Funcionamento geral:** obtêm-se os três números do usuário; realiza-se multiplicação e depois a divisão; mostrar o resultado para o usuário.

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe21.por**

❖ Exercício 21 – Média aritmética de 3 notas

1. Faça um algoritmo que leia 3 notas (reais) de provas realizadas por um aluno. Mostre a sua média aritmética final.
- **Objetivo:** realizar a soma de três números reais, achar a média dos três valores (dividir por 3) e apresentar o resultado.
 - **Comentário:** como existem três números, cada um deve estar armazenado em um local da memória, então devem ser criadas três variáveis para armazenar cada um deles. Além disso, deve-se criar uma variável para armazenar o resultado.
 - **Funcionamento geral:** obtêm as três notas do usuário; realiza-se a divisão; mostra o resultado para o usuário.

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe22.por**

❖ Exercício 22 – Metros, centímetros e milímetros

1. Escreva um programa que leia um valor em metros e o exiba convertido em centímetros e milímetros.
- **Objetivo:** obter um número inteiro do usuário, mostrar o resultado deste número convertido em centímetros e milímetros.
 - **Comentário:** como será lido apenas um número e realizado um cálculo com este, então será necessária uma variável para armazenar o número lido e outras duas para cada resultado.
 - **Funcionamento geral:** obtém o valor do usuário; multiplica-se por 100 para centímetros e multiplica-se por 1000 para milímetros; mostrar o resultado para o usuário.

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe23.por**

❖ Exercício 23 – Calcular o valor de uma prestação

1. A Loja Mamão com Açúcar está vendendo seus produtos em 12 (doze) prestações sem juros. Faça um algoritmo que receba um valor de uma compra e mostre o valor das prestações em 12 vezes.
- **Objetivo:** Obter o valor total do produto e calcular o valor das prestações. Mostrar o valor das prestações ao final.
 - **Comentário:** o objetivo é obter o valor total do item, realizar a divisão por 12 e mostrar o valor da prestação ao usuário.
 - **Funcionamento geral:** obtêm o valor do item, armazenar em uma variável do tipo real. Efetuar o cálculo, dividindo o valor de item em 12 vezes e mostrar o valor da prestação ao usuário.

Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos
- Salve o arquivo utilizando o seguinte padrão: **exe24.por**

❖ Exercício 24 – Agenda de alunos

1. Faça um algoritmo que peça o **nome do aluno**, **código do aluno (prontuário)**, **telefone** e **e-mail** e mostre estas informações na tela
- **Objetivo:** obter as seguintes informações de um aluno : nome , prontuário, telefone e e-mail. Armazenar estas informações em variáveis correspondentes e mostrar todos os dados coletados ao final.
 - **Comentário:** o objetivo é obter diversos tipos de dados diferentes, como nome (caractere), prontuário (inteiro), telefone (inteiro), e-mail (caractere) armazenar os dados em variáveis que aceitem o tipo de dado correspondente e mostrar os valores no final.
 - **Funcionamento geral:** obtêm os valores do aluno; salvar em variáveis compatíveis com cada tipo de dado e mostrar o resultado para o usuário.

Exercícios

❖ Exercício 25 – Calcular a expressão: $D=(R+S)/4$

1. Escreva um algoritmo que leia três números reais positivos (**A**, **B**, **C**).
2. Calcule e mostre o resultado da seguinte expressão:

$$D = (R+S)/4$$

Onde: $R = (A+B)*2$ e $S = (B+C)*A$

- **Objetivo:** obter três números do usuário (A,B,C), calcular o valor de D, conforme a equação, e mostrar o resultado para o usuário.
- **Comentário:** o objetivo final é calcular o valor de D. Porém, ao analisar a equação é possível notar que primeiro deve-se encontrar o valor de R e S. Então, o algoritmo deve primeiro resolver R e S, e somente depois D. Será necessário uma variável para cada valor que será necessário armazenar. Como o enunciado deixou claro que o resultado deve mostrar as casas decimais, então as variáveis deverão ser do tipo Reais.
- **Funcionamento geral:** obtêm os três valores do usuário (A,B,C); calcula-se R e S; calcula-se D; mostra-se o resultado para o usuário.

Biblioteca de Matemática

- Um conjunto de funções e comandos recebe o nome de **Biblioteca**.
- Existem diversos tipos de bibliotecas, cada uma com funções para atender a determinados problemas.
- Para se utilizar uma biblioteca é necessário primeiro **importa-la** para o seu programa.
- Para importar a biblioteca de matemática, insira o comando abaixo no início do programa:
- **inclua biblioteca Matematica**

```
1 // exibir o número PI com 4 casas decimais
2 programa
3 {
4     inclua biblioteca Matematica --> mat
5     funcao inicio()
6     {
7         real resultado
8         resultado = mat.arredondar(mat.PI,4)
9         escreva (resultado)
10    }
11 }
```

Funções da Biblioteca de Matemática

- A biblioteca de matemática permite realizar diversas funções como:

1. **Arredondar um número** para 2 casas decimais, por exemplo

arredondamento = **mat.arredondar**(numero, 2)

2. **Funções trigonométricas**

seno = **mat.seno**(numero)

cosseno = **mat.cosseno**(numero)

tangente = **mat.tangente**(numero)

3. **Calcular a Potência de um número**

cubo = **mat.potencia**(base, 3.0)

4. **Calcular a Raiz Quadrada de um número**

raiz = **mat.raiz**(numero, 2.0)

5. **Mostrar o valor do número PI**

escreva("O valor de PI é: ", **mat.PI**)

Exercícios

❖ Exercício 26 – Consumo de um automóvel

1. Escrever um algoritmo para determinar o consumo médio de um automóvel sendo fornecida a distância total percorrida pelo automóvel (km) e o total de combustível gasto (litros) para percorrer esta distancia.
2. Consumo é dado em quilômetros por litro (km/l).



❖ Exercício 27 – Média de gols do campeonato

1. Escreva um algoritmo que receba o número “np” de partidas do campeonato brasileiro e o número “ng” de gols marcados em todo o campeonato.
2. Calcule e mostre a média de gols por partida, sendo: que a média de gols = número de gols/número de partidas).



❖ Exercício 28 – Aumento de salário

1. Faça um algoritmo para ler o **nome** e **salário** de um funcionário e calcular um aumento de **25%**. Após aplicado o aumento, desconte **5%** de impostos.
2. Mostre o nome do funcionário, o salário inicial, o salário com aumento e o salário final.

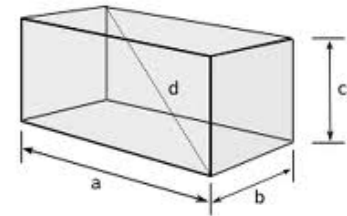
Exercícios

❖ Exercício 29 – As operações matemáticas

1. Faça um algoritmo que receba dois números e ao final mostre a soma, subtração, multiplicação, divisão, potenciação e raiz dos números lidos.

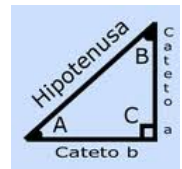
❖ Exercício 30 – Diagonal do Paralelepípedo

1. Entrar com os lados a, b e c de um paralelepípedo.
2. Calcular e mostrar a diagonal.



❖ Exercício 31 – Hipotenusa do triângulo

1. Entre com os lados do cateto de um triângulo retângulo e mostre a hipotenusa.



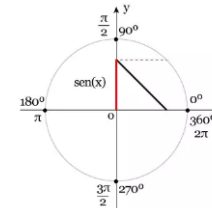
❖ Exercício 32 – Conta do restaurante

1. Todo restaurante, cobra 10% para o garçom. Faça um algoritmo que leia o valor gasto com despesas realizadas por “ x ” clientes em um restaurante, divida a conta pela quantidade “ x ” de clientes na mesa e imprima o valor a ser pago por pessoa com a gorjeta do garçom.

Exercícios

❖ Exercício 33 – As operações trigonométricas

1. Faça um programa que leia um ângulo qualquer e mostre na tela o valor do seno, cosseno e tangente desse ângulo.



❖ Exercício 34 – Carro alugado

1. Escreva um programa que pergunte a quantidade de Km percorridos por um carro alugado e a quantidade de dias pelos quais ele foi alugado.
2. Calcule o preço a pagar, sabendo que o carro custa R\$ 60 por dia e R\$ 0,15 por Km rodado.



❖ Exercício 35 – Separando os dígitos

1. Faça um programa que leia um número de 4 dígitos (ex: 1492) e mostre na tela cada um dos dígitos separados

2. Exemplo: Digite um número: 1492

milhar: 1 centena: 4 dezena: 9 unidade: 2

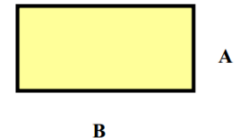


Exercícios

- Utilize o programa **Portugol Studio** para elaborar os algoritmos:

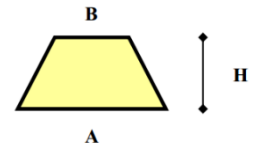
❖ Exercício 36 – Área do retângulo.

- Fazer um algoritmo que solicite o valor dos lados **a** e **b** de um retângulo e calcule a sua área.



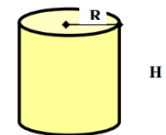
❖ Exercício 37 – Área do trapézio

- Fazer um algoritmo que solicite o valor das bases **a**, **b** e da altura **h** de um trapézio e calcule a sua área.



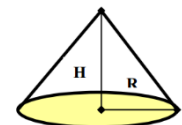
❖ Exercício 38 – Volume do cilindro

- Fazer um algoritmo que solicite o valor do raio **r** e a altura **h** de um cilindro e calcule o seu volume.



❖ Exercício 39 – Volume do cone

- Fazer um algoritmo que solicite o valor do raio **r** e a altura **h** de um cone e calcule o seu volume.



- Utilize o programa **Portugol Studio** para elaborar os algoritmos:

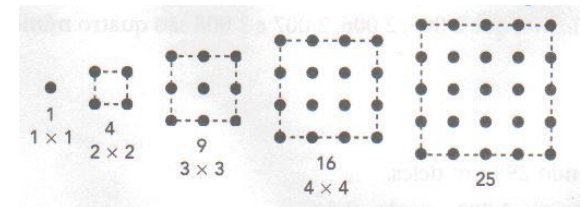
❖ Exercício 40 – Conversão de dólar para real

1. Pesquisar o valor da cotação do dólar atual.
2. Informar a quantidade de dólares que deseja converter
3. Converter para reais e mostrar a informação.
4. Apresentar o resultado da conversão



❖ Exercício 41 – Soma dos quadrados dos números

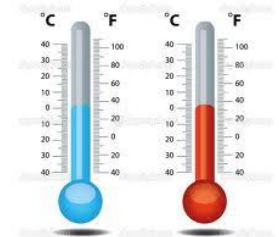
1. Informar o valor de 4 números quaisquer.
2. Elevar ao quadrado cada um destes números.
3. Somar o valor dos quadrados dos 4 números.
4. Apresentar o resultado final da soma dos quadrados.



- Utilize o programa **Portugol Studio** para elaborar os algoritmos:

❖ Exercício 42 – Conversão de temperatura

- Informar uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit.
- A fórmula de conversão é: $F = (9 * C / 5) + 32$, sendo **F** a temperatura em Fahrenheit e **C** a temperatura em Celsius.



❖ Exercício 43 – Salário líquido

- Faça um algoritmo que solicite o nome e o **salário bruto (sem impostos)** de um funcionário de uma empresa.
- Sabe-se que o imposto de renda é de 27% sobre o salário.
- Calcule e mostre o **salário líquido (sem impostos)**
- Ao final informe o nome e o salário líquido do funcionário.



Exercícios

- Utilize o programa **Portugol Studio** para elaborar o algoritmo:
- ❖ **Exercício 44 – Comissão de vendedoras de roupas**
- Construa um algoritmo para calcular o valor da comissão de uma vendedora de roupas.
- A sua comissão será de **12%** do total de vendas realizado
- Sabe-se que a loja vende três tipos de roupa: **calça, blusa e saia**.
- Você deve informar os seguintes dados (variáveis):
 1. **Nome** da vendedora
 2. **Preço unitário** de cada roupa (calça, blusa, saia)
 3. **Quantidade** vendida de cada roupa (calça, blusa, saia)
- Ao final mostre o **nome da vendedora** e a **comissão da vendedora**.



Exercícios

- ❖ **Exercício 45 – Custo da pintura de uma sala**
 - Construa um algoritmo para calcular o **custo da pintura** de uma sala.
 - Considere que a sala não possui portas ou janelas.
 - Para efetuar este cálculo você precisa coletar os seguintes dados:
 1. **Medidas da sala:** largura, comprimento e altura das paredes.
 2. **Quantidade de demão utilizada na pintura.**
 3. **Rendimento de uma lata de tinta por metro quadrado.** (pesquise !)
 4. **Custo da lata de tinta.** (pesquise !)
 5. **Custo de mão de obra do pintor por m².**
 - Calcule e mostre:
 1. **Quantidade de latas de tinta.**
 2. **O custo da total gasto com a tinta e custo da mão de obra do pintor.**
 3. **O custo total da obra (pintura) a ser executada.**

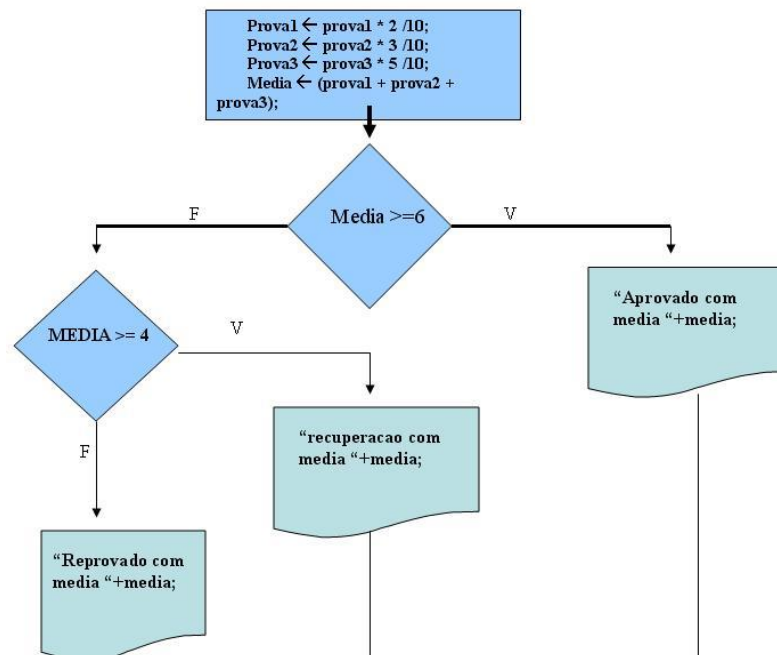


Aula 5 – Algoritmos Condicionais

```
HELLO, WORLD!  
  
LIST  
10 HOME  
20 INVERSE  
30 PRINT "HELLO, WORLD!"  
40 NORMAL  
50 PRINT CHR# (7)  
■
```

Algoritmos Condicionais

- **Algoritmo Condicional**
- Na vida real tomamos decisões a todo o momento baseadas em uma situação existente.
- Em um algoritmo, chamamos esta situação de **condição**.
- Associada a uma condição, existirá uma alternativa possível de ações.



Algoritmos Condicionais

- Analise a frase abaixo:
- "se eu tiver R\$ 10,00 sobrando então irei ao cinema hoje à noite."
- A condição nesta frase é "**ter R\$ 10,00 sobrando**".
- Ela é uma expressão lógica, pois a pergunta "Tenho R\$ 10,00 sobrando?" Deve ser respondida com "**Sim**" ou "**Não**".
- Lembre-se, então: em um algoritmo, toda condição tem que ser uma expressão lógica, algo que possa-se pensar como "isto é **VERDADEIRO**" ou "isto é **FALSO**".
- Se a condição for verdadeira, a ação a ser executada é "irei ao cinema". Então, em um algoritmo, as ações são um ou mais comandos que serão realizados apenas se a avaliação da condição resulta **VERDADEIRO**.

Algoritmos Condicionais

- Vamos colocar agora a frase do exemplo anterior em outra forma, mais parecida com nosso Português Estruturado:

se “eu tiver R\$ 10,00” então

“Irei ao cinema”

Como próximo passo, vamos generalizar a estrutura que criamos acima:

```
1  programa
2  {
3      funcao inicio ()
4      {
5          se (dinheiro > 10)
6          {
7              escreva("Irei ao cinema")
8          }
9      }
10 }
```

Algoritmos Condicionais

- O exemplo citado poderia ser estendido para o caso do sujeito não ter dinheiro sobrando:
- **se** tiver R\$ 10,00 “Irei ao cinema”,
- mas **senao** tiver “Ficarei vendo TV em casa”.
- Neste caso, uma codificação possível para esse algoritmo seria:

```
1 programa
2 {
3     funcao inicio ()
4     {
5         se (dinheiro > 10)
6         {
7             escreva("Irei ao cinema")
8         }
9         senao
10        {
11            escreva("Ficarei vendo TV em casa")
12        }
13    }
14 }
```

Condicional Simples

- A condicional simples analisa apenas **UMA** condição e exibe um resultado em que esta condição é **VERDADEIRA**
- Condição para votar nas eleições:
- Se você tiver mais que 16 anos, você está apto para votar

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro idade
6          escreva ("Digite sua idade:\n")
7          leia (idade)
8          limpa()
9          se (idade >= 16)
10         {
11             escreva ("Você está apto a votar.")
12         }
13     }
14 }
```

Condicional Composta

- A condicional composta analisa as duas possibilidades de uma condição e exibe um resultado em que esta condição é **VERDADEIRA**, mas também exibe um resultado em que esta condição é **FALSA**.
- Condição para votar nas eleições:
- Se você tiver mais que 16 anos, você está apto a votar
- Senão, você ainda não tem idade para votar

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro idade
6          escreva ("Digite sua idade:\n")
7          leia (idade)
8          limpa ()
9          se (idade >= 16)
10         {
11             escreva ("Você está apto a votar.")
12         }
13         senao
14         {
15             escreva ("Você ainda não tem idade para votar !")
16         }
17     }
18 }
```

Múltiplas Condicionais

- São variantes de uma condição simples, analisada por diferentes aspectos
- Se você tiver **idade menor que 16 anos**, você não pode votar
- Se você tiver **idade maior ou igual a 16 anos**, você está apto a votar
- Se você tiver **idade maior que 70 anos**, você não precisa mais votar

```
1  programa
2  {
3      funcao inicio ()
4      {
5          inteiro idade=20
6          se (idade < 16)
7          {
8              escreva("Você não pode votar ainda !")
9          }
10         se (idade >= 16 e idade < 70)
11         {
12             escreva("Você está apto para votar !")
13         }
14         se (idade >= 70)
15         {
16             escreva("Você não precisa mais votar !")
17         }
18     }
19 }
```


Condicional Aninhada

- São subcondições ou condições dentro de outras condições
- Se você tiver **idade menor que 16 anos**, você não pode votar
- Se você tiver **idade maior ou igual a 16 anos**, você está apto a votar
- Se você tiver **idade maior que 70 anos**, você não precisa mais votar

```
1  programa
2  {
3      funcao inicio ()
4      {
5          inteiro idade=70
6          se (idade < 16)
7          {
8              escreva("Você não pode votar ainda !")
9          }
10         se (idade >= 16)
11         {
12             escreva("Você está apto para votar !\n")
13             se (idade >= 70)
14             {
15                 escreva("Mas para você, o voto não é mais obrigatório !")
16             }
17         }
18     }
19 }
```

Operadores Relacionais

Os operadores relacionais são: **== != > < >= <=**

Os operadores relacionais só podem ter como resultado um valor booleano, que pode assumir apenas os valores verdadeiro ou falso.

Veja exemplos de cada uma dessas operações:

igual: ==

diferente: !=

maior que: >

menor que: <

maior ou igual a: >=

menor ou igual a: <=

Operadores Lógicos

Os operadores lógicos são: **e ou nao**

Operadores lógicos servem para conectar duas expressões ou para negar uma expressão. Os operadores lógicos mais usados são:

operador e: e

operador ou: ou

operador não: nao

exemplo: você poderia querer verificar se um aluno teve nota no ENEM superior a 700 **e** nota em Matemática maior que 800 **e** nota em Redação de pelo menos 650:

```
1  programa
2  {
3      funcao inicio()
4      {
5          se (notaEnem > 700 e notaMatematica > 800 e notaRedacao >= 650)
6          {
7              escreva("Aprovado!\n")
8          }
9      }
10 }
```

Exercícios

❖ Exercício 46 – Maior que 10 ?

1. Faça um algoritmo que receba um número e mostre uma mensagem caso este número seja igual, maior ou menor que 10.

❖ Exercício 47 – Valor positivo para o raio do círculo !

1. Faça um algoritmo para calcular a área de um círculo, fornecido o valor do raio, que deve ser positivo. Se o raio informado for negativo ou zero, mostrar mensagem que o valor do raio é inválido.

❖ Exercício 48 – Positivo ou Negativo ?

1. Faça um algoritmo que receba um número “N” qualquer.
2. Mostre se o número é positivo, negativo ou zero.

❖ Exercício 49 – Iguais ou diferentes ?

1. Faça um algoritmo que leia dois números e identifique se são iguais ou diferentes. Caso eles sejam iguais imprima uma mensagem dizendo que eles são iguais. Caso sejam diferentes, informe qual número é o maior, e uma mensagem que são diferentes.

Exercícios

❖ Exercício 50 – Meses

1. Faça um algoritmo que leia um número que represente um determinado mês do ano. Após a leitura escreva por extenso qual foi o mês lido.
2. Caso o número digitado não esteja na faixa de 1..12 escreva uma mensagem informando o usuário do erro da digitação.

❖ Exercício 51 – $(A+B) < C$

1. Faça um algoritmo que leia os valores A, B, C e diga se a soma de $A + B$ é menor que C.

❖ Exercício 52 – $(A+B)$ ou $(A*B)$

1. Faça um algoritmo que leia dois valores inteiros A e B. Se os valores forem iguais deverá somar os dois, caso contrário multiplique A por B.

❖ Exercício 53 – Sucessor

1. Crie um algoritmo que leia um número entre 0 e 60 e escreva o seu sucessor, sabendo que o sucessor de 60 é 0. Para qualquer número diferente deste intervalo informar mensagem “número inválido”.

Exercícios

❖ Exercício 54 – Calculadora

1. Escreva um algoritmo para simular uma calculadora.
2. A calculadora deve ler 2 números e executar somente as operações básicas soma (+), subtração (-), multiplicação (*) e divisão (/),
3. Qualquer operação diferente dessas deve receber como resposta a mensagem “Opção não disponível”, do contrário, a resposta deve ser o resultado da operação.

❖ Exercício 55 – Entre 100 e 200

1. Faça um algoritmo que receba um número e mostre uma mensagem se este número está no intervalo entre 100 e 200, inclusive.
2. Mostre uma mensagem se este número for maior que 200.
3. Mostre uma mensagem se este número for menor que 100.

❖ Exercício 56 – Par ou ímpar ?

1. Faça um algoritmo que leia um número N e verifique se ele é par ou ímpar. Escreva o número e informe se é par ou ímpar.

Exercícios

❖ Exercício 57 – Maior valor

1. Escreva um algoritmo que leia 3 números inteiros (x,y,z) e mostre o maior entre eles. A comparação entre os números deve ser por pares.

❖ Exercício 58 – Números Múltiplos

1. Elaborar um algoritmo que leia dois números inteiros **N1** e **N2**.
2. Verifique se o número N1 é múltiplo de N2 e escreva a mensagem correspondente: “N1 é múltiplo de N2” ou “N1 não é múltiplo de N2”.

❖ Exercício 59 – Serviço militar

1. Escrever um algoritmo que leia os dados de uma pessoa (nome, sexo, idade altura e saúde) e informe se esta pessoa está apta ou não para cumprir o serviço militar obrigatório.
2. Condição para prestar serviço militar: ser homem, idade igual a 18 anos, altura mínima 1,70cm de altura, além de gozar de boa saúde.
3. Informar nome seguido de mensagem informando se está apto ou não.

Exercícios

❖ Exercício 60 – Aprovado ou reprovado?

1. Dado um algoritmo que leia o nome e as quatro notas bimestrais de um aluno, calcule a média aritmética das notas deste aluno.
2. Mostre o nome, a média calculada e seguinte mensagem:
 - “**Aprovado**” se Média ≥ 6.0
 - “**Recuperação**”, se $3.0 \leq \text{Média} < 6.0$
 - “**Reprovado**”, se Média < 3.0

❖ Exercício 61 – Nadadores

1. Elabore um algoritmo que dada a idade de um nadador classifica-o em uma das seguintes categorias:
 - **infantil A** = 5 - 7 anos
 - **infantil B** = 8 - 10 anos
 - **juvenil A** = 11 - 13 anos
 - **juvenil B** = 14 - 17 anos
 - **adulto** = maiores de 18 anos

Exercícios

❖ Exercício 62 – Promoção para Carango Velho

1. A concessionária de veículos “**CARANGO VELHO**” está vendendo os seus veículos com desconto. Faça um algoritmo que calcule e exiba o valor do desconto e o valor a ser pago pelo cliente. O desconto deverá ser calculado de acordo com o ano do veículo:
2. Até 2010 (35%), de 2011 à 2015 (25%) e de 2016 à 2020 (15%).

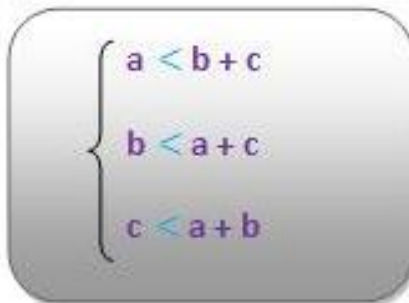
❖ Exercício 63 – Aumento de salário

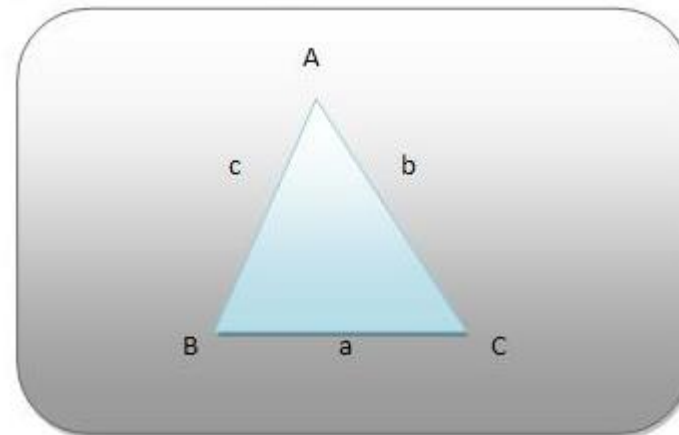
1. Pesquisar o salário mínimo atual.
2. Escrever um algoritmo para uma empresa que decide dar um reajuste de salário a seus funcionários de acordo com os seguintes critérios:
 - 50% para aqueles que ganham menos do que três salários mínimos;
 - 20% para aqueles que ganham entre três até dez salários mínimos;
 - 15% para aqueles que ganham acima de dez até vinte salários mínimos;
 - 10% para os demais funcionários.
3. Mostrar nome do funcionário, o salário atual e o salário com o reajuste.

Exercícios

❖ Exercício 64 – Triângulo ?

1. Elaborar um algoritmo que leia 3 valores a,b,c, que representam os lados de um triângulo. Supor que os valores lidos são inteiros e positivos.
2. Verifique se eles formam ou não um triângulo. Informar isto através de uma mensagem. Ex: “*Não formam um triângulo*”
3. Caso os valores formem um triângulo, informar e verificar qual é o tipo de triângulo formado: equilátero, isósceles ou escaleno.
4. Obs importante: Para construir um triângulo é necessário que a medida de qualquer um dos lados seja menor que a soma das medidas dos outros dois.
5. Exemplo:


$$\begin{cases} a < b + c \\ b < a + c \\ c < a + b \end{cases}$$



Aula 6 – Algoritmos de Repetição

```
HELLO, WORLD!  
  
LIST  
10 HOME  
20 INVERSE  
30 PRINT "HELLO, WORLD!"  
40 NORMAL  
50 PRINT CHR$ (7)  
60
```

Algoritmos de Repetição

- **Algoritmos de repetição ou laços de repetição**
- São algoritmos que executam uma função e repetem uma sequência de comandos um determinado número de vezes, ou enquanto uma condição pré-determinada ou pós-determinada existir.
- Estes algoritmos podem ser resolvidos com a utilização de laços de repetição. Um laço de repetição, como sugere o próprio nome, é um comando onde uma quantidade de comandos se repete até que uma determinada condição seja verdadeira.
- O Portugol contém 3 tipos de laços de repetição: laço com variável de controle, pré-testado, pós-testado
- Nesta seção, serão abordados os seguintes tópicos:
 1. **Para**
 2. **Enquanto**
 3. **Faça-enquanto**

Para

- A maioria das linguagens de programação suportam o **para**. Ele define três ações em uma sintaxe compacta. Essas ações são inicialização, condição e atualização.
- A **inicialização** é usada para inicializar variáveis e é a primeira coisa que é feita no para. Só é feita no começo. A **condição** é feita antes de cada iteração, inclusive a primeira. A **atualização** é feita após cada iteração.
- O uso mais comum de um para é inicializar uma variável, que é chamada de variável de controle, testá-la na condição do for e depois atualizá-la. Por causa do uso da variável de controle, esse tipo de repetição é chamado também de repetição com variável de controle.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro i
6          para (i = 1; i <= 10; i++)
7          {
8              escreva(i, "\n")
9          }
10     }
11 }
```

Exemplos

- **Exemplo 1 – Fazer uma pergunta 10 vezes**
- Este exemplo o programa solicita ao usuário que informe a temperatura no local 10 vezes seguidas.
- Este é um exemplo de repetição simples

```
1  programa
2  {
3      inclua biblioteca Util --> u
4      funcao inicio()
5      {
6          inteiro temp,i
7          para (i=1;i<=10;i++)
8          {
9              escreva ("Qual a temperatura agora ?\n")
10             leia (temp)
11             escreva ("Obrigado por informar !")
12             u.aguarde(2500)
13             limpa()
14         }
15     }
16 }
```

Exemplos

- **Exemplo 2 - Fazer a tabuada de um número qualquer**
- Este exemplo pede ao usuário que informe um número inteiro. Logo após, utiliza um laço de repetição do tipo "para", para calcular e exibir a tabuada do número informado.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro numero, resultado, contador
6
7          escreva("Informe um número para ver sua tabuada: ")
8          leia(numero)
9
10         limpa()
11
12         para (contador = 1; contador <= 10; contador++)
13         {
14             resultado = numero * contador
15             escreva (numero, " X ", contador, " = ", resultado , "\n")
16         }
17     }
18 }
```

Contadores

- Uma das funções mais importantes inseridas dentro de uma estrutura de repetição é a de um **contador**.
- O contador analisa uma quantidade de vezes que uma certa situação é executada dentro de um algoritmo.
- Os contadores são declarados como sendo do tipo inteiro.
- No exemplo abaixo, o papel do **i** é a de ser o **CONTADOR** de vezes que a repetição PARA será executada.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro i
6          para (i = 1; i <= 10; i++)
7          {
8              escreva("Agora o contador é ",i,".\n")
9          }
10     }
11 }
```


Exemplos

1. Faça um algoritmo que leia a idade de 5 pessoas.
2. Verifique quantas pessoas são maiores ou menores de idade e mostre o resultado ao final.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro x,idade,maior=0,menor=0
6          para (x=1;x<=5;x++)
7          {
8              limpa()
9              escreva ("Informe sua idade:\n")
10             leia (idade)
11             se (idade>=18)
12             {
13                 maior++
14             }
15             senao
16             {
17                 menor++
18             }
19         }
20         limpa()
21         escreva ("Neste grupo temos ",maior," maiores de idade.\n")
22         escreva ("Neste grupo temos ",menor," menores de idade.\n")
23     }
24 }
```

Exemplos

1. Escrever um algoritmo que leia o sexo de 10 pessoas.
2. No final informe total de homens e de mulheres.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro i,mulher=0,homem=0
6          caracter sexo
7          para (i=1;i<=10;i++)
8          {
9              limpa()
10             escreva ("Informe seu sexo:\n")
11             escreva ("Digite M para masculino e F para feminino.\n")
12             leia (sexo)
13             se (sexo=='M')
14             {
15                 homem++
16             }
17             senao
18             {
19                 mulher++
20             }
21         }
22         limpa()
23         escreva ("Neste grupo temos ",homem," pessoas do sexo masculino.\n")
24         escreva ("Neste grupo temos ",mulher," pessoas do sexo feminino.\n")
25     }
26 }
```

Exemplos

1. Ler 80 números quaisquer e ao final informar quantos números estão no intervalo entre 10 (inclusive) e 150 (inclusive).
2. Utilize o operador **E** neste exercício, para concatenar 2 condições.

```
1  programa
2  {
3      inclua biblioteca Util --> u
4      funcao inicio()
5      {
6          inteiro x,num,inter=0
7          para (x=0;x<=80;x++)
8          {
9              num=u.sorteia(0,1000)
10             escreva ("Número aleatório = ",num,".\n")
11             u.aguarde(500)
12             se (num>=10 e num<=150)
13             {
14                 inter++
15             }
16         }
17         limpa()
18         escreva ("Foram digitados ",inter," números dentro do intervalo.\n")
19     }
20 }
```

Incrementadores

- Outro tipo de variáveis muito utilizada em repetição são as variáveis que chamamos de **INCREMENTADORES**.
- Estas são variáveis que exercem o papel de **incrementar** um valor da mesma se alguma determinada condição for atendida no código.
- Em geral os incrementadores são declarados como sendo do tipo **inteiro**.
- A ação de incrementar é uma funcionalidade matemática que se realiza entre variáveis do programa e servem basicamente para incrementar (acrescentar ou somar) um valor a uma variável .
- Exemplo:
- A variável **número** exerce o papel de um valor qualquer.
- A variável **soma** exerce o papel de incrementador.
- Abaixo temos a **atribuição** do resultado da soma entre **numero + soma** para a variável soma:
- **soma = soma + numero**
- Na expressão acima leia-se: soma recebe o valor de número + soma

Exemplos

1. Escrever um algoritmo que leia a nota de 40 alunos e calcule a média da turma.
2. Obs: a função da variável **SOMA** é somar todas as 40 notas dos alunos, para depois ser calculada a média.

```
1  programa
2  {
3      inclua biblioteca Util --> u
4      funcao inicio()
5      {
6          inteiro x,nota,soma=0
7          real media
8          para (x=0;x<=40;x++)
9          {
10             nota=u.sorteia(0,10)
11             escreva ("Nota do aluno ",x," é ",nota,".\n")
12             u.aguarde(500)
13             soma = soma+nota
14         }
15         media = soma/40
16         limpa()
17         escreva ("A média da turma é ",media,".\n")
18     }
19 }
```

Exercícios

❖ Exercício 65

1. Escrever um algoritmo que leia 50 valores inteiros quaisquer e conte quantos destes valores são negativos, escrevendo ao final esta informação.

❖ Exercício 66

1. Escrever um algoritmo que leia 500 valores e conte quantos destes valores são números pares e quantos são números ímpares, escrevendo esta informação.

❖ Exercício 67

1. Em uma loja de automóveis, cada carro à venda tem uma ficha contendo a marca, modelo e a cor do carro.
2. Dado uma relação de 16 carros, mostre a quantidade de carros da cor azul.

❖ Exercício 68

1. Dado uma série de 1500 valores reais, faça um algoritmo que calcule e escreva a média aritmética destes valores.

Exercícios

❖ Exercício 69

1. O professor deseja calcular a média de uma turma de 25 alunos.
2. Para isso solicite a nota final de cada aluno e calcule a média aritmética desta turma. Ao final verifique o rendimento desta turma:
3. Se média da turma for superior a 8 escreva: “Esta turma é excelente !!!”.
4. Se média da turma for entre 5 e 8 escreva: “Esta turma é boa.”.
5. Se média da turma for inferior a 4 escreva: “Esta turma já era...”.

❖ Exercício 70

1. Faça um algoritmo para somar todos os números pares de 0 a 10.000 e ao final mostre o resultado desta soma.

❖ Exercício 71

1. Suponha que para cada aluno de sua sala exista uma ficha contendo o nome e a idade do aluno.
2. Supondo que existam 50 alunos, faça um algoritmo que determine quantos alunos tem idade maior que 15.

Exercícios

❖ Exercício 72

1. Escrever um algoritmo que mostre todos os números ímpares que sejam múltiplos de 3 entre 10 e 2000.

❖ Exercício 73

- Joãozinho nasceu com 37.6 centímetros e sua mãe verificou que ele cresce 6.75 centímetros em média por ano.
- Que altura terá Joãozinho com 20 anos de idade ?

❖ Exercício 74

- Escreva um algoritmo que leia 500 valores inteiros e positivos e:
 - a) encontre o maior valor;
 - b) encontre o menor valor;
 - c) calcule a soma de todos os valores lidos;
 - d) calcule a média dos números lidos.

Exercícios

❖ Exercício 75

1. Faça um algoritmo que receba o preço de custo e o preço de venda de 40 produtos.
2. Mostre como resultado se houve lucro, prejuízo ou empate para cada produto. Informe media de preço de custo e do preço de venda.

❖ Exercício 76

1. Escrever um algoritmo que leia 20 valores para uma variável n e, para cada um deles, calcule a tabuada de 1 até n . Mostre a tabuada na forma:
 - $1 \times n = n$ $2 \times n = 2n$ $3 \times n = 3n$

❖ Exercício 77

- Ler 2 valores quaisquer. Considere que o segundo valor lido será sempre maior que o primeiro valor lido.
- Calcular a soma de todos os números inteiros existentes entre estes 2 valores lidos (incluindo os próprios valores).

Enquanto

- **Algoritmos de repetição ou laços de repetição**
- Se fosse necessário a elaboração de um jogo, como por exemplo um jogo da velha, e enquanto houvessem lugares disponíveis no tabuleiro, este jogo devesse continuar, como faríamos para que o algoritmo tivesse este comportamento? É simples. O comando **enquanto** poderia fazer esse teste lógico. A função do comando enquanto é: executar uma lista de comandos enquanto uma determinada condição for verdadeira.
- A sintaxe é respectivamente a palavra reservada enquanto, a condição a ser testada entre parênteses, e entre chaves a lista de instruções que se deseja executar.
- Verifique o exemplo abaixo:

```
1  {  
2  
3  {  
4      logico condicao = verdadeiro  
5      enquanto (condicao)  
6      {  
7          //Executa a as instruções dentro do laço enquanto a condicao for verdadeira  
8      }  
9  }  
10 }
```

Enquanto

- Verifique o exemplo abaixo:
- O algoritmo abaixo verifica uma variável do tipo carácter. Enquanto a variável for diferente da letra 'S' o comando enquanto será executado, assim como as instruções dentro dele. No momento em que o usuário atribuir 'S' a variável, o comando enquanto terminará e o programa chega ao seu final.

```
1  programa
2  {
3      funcao inicio()
4      {
5          caracter parar
6          parar = 'N'
7
8          enquanto (parar != 'S')
9          {
10             escreva ("Iniciando um laço de repetição...\n")
11             escreva ("Deseja parar o laço? (S/N)\n")
12             leia (parar)
13         }
14     }
15 }
```

Enquanto

- Um loop infinito é o mesmo que uma “repetição infinita”.
- Na área de informática e programação de softwares, por exemplo, pode representar um erro na execução de determinado programa, quando este passa a seguir repetidamente a mesma sequência de instruções.
- **O loop infinito deve ser evitado na programação e geralmente está associado a um erro de conceito ou de programação.**
- Verifique o exemplo abaixo e analise que o programa não possui uma condição de encerramento do laço de repetição.

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro loop = 1
6          enquanto (loop == 1)
7              escreva("Este programa é um loop infinito !!!\n")
8      }
9  }
```

Exemplos

- **Exemplo 1 - Fazer uma contagem regressiva**
- Este exemplo utiliza um laço de repetição e uma variável para exibir uma contagem regressiva na tela.

```
1  programa
2  {
3      inclua biblioteca Util
4
5      funcao inicio()
6      {
7          inteiro contador = 10
8
9          enquanto (contador > 0)
10         {
11             limpa()
12             escreva ("Detonação em: ", contador)
13
14             contador = contador - 1
15             Util.aguarde(1000) // Aguarda 1000 milisegundos (1 segundo)
16         }
17
18         limpa()
19         escreva ("Booom!\n")
20     }
21 }
```

Exemplos

- Exemplo 2 – Acender a luz

- Escrever um algoritmo faça um programa que simule um interruptor de luz elétrico utilizando a estrutura de repetição enquanto.

```
1  programa
2  {
3      inclua biblioteca Util
4      funcao inicio()
5      {
6          inteiro luz
7          escreva ("Clique 1 para acender a luz.\n")
8          leia (luz)
9          limpa()
10         enquanto (luz==1)
11         {
12             escreva ("Luz Acesa.\n")
13             Util.aguarde(3000) // Aguarda 3000 milisegundos (3 segundos)
14             escreva ("Deseja apagar a luz ?\n")
15             escreva ("Clique 0 para apagar ou 1 para manter acesa.\n")
16             leia (luz)
17             limpa()
18         }
19         escreva ("Luz apagada.\n")
20     }
21 }
```

Exemplos

- Exemplo 3 – Produzindo parafusos

1. Escrever um algoritmo que faça uma parafusos até que o insumo (aço) se acabe. A maquina tem capacidade de 10kg de aço e cada parafuso consome 25g de aço para ser produzido.
2. Verifique quantos parafusos foram produzidos ao final do processo.

```
1  programa
2  {
3      inclua biblioteca Util
4      funcao inicio()
5      {
6          inteiro parafuso=0
7          real aco=10
8          enquanto (aco >= 0.025)
9          {
10             parafuso++
11             escreva ("Produzindo parafuso nº ",parafuso,"...\n")
12             Util.aguarde(50) // Aguarda 50 milisegundos (0,05 segundos)
13             aco = aco - 0.025
14             limpa()
15         }
16         escreva ("Total de parafusos produzidos = ",parafuso,"\n")
17     }
18 }
```

Exemplos

- **Exemplo 4 – Limite de ebulição da água (100° Celsius)**
 1. Escrever um algoritmo que verifique, através de sensores de temperatura, se a água ainda está líquida ou está em estado de ebulição.
 2. Simule que a temperatura aumenta um grau a cada ciclo de medição, iniciando por zero graus. Escreva uma mensagem para apresentar o estado da água.

```
1  programa
2  {
3      inclua biblioteca Util
4      funcao inicio()
5      {
6          inteiro temp=0
7          enquanto (temp <= 100)
8          {
9              escreva ("Temperatura da água = ",temp," °C.\n")
10             escreva ("Água em estado líquido...\n")
11             Util.aguarde(500) // Aguarda 500 milisegundos (0,5 segundos)
12             temp++
13             limpa()
14         }
15         escreva ("Água em ebulição !!!.\n")
16     }
17 }
```


Exemplos

- **Exemplo 4 – Utilizando uma variável como condição inicial**
- Escrever um algoritmo calcule a área de um quadrado e que ao final pergunte se deseja realizar o cálculo novamente.

```
1  programa
2  {
3      inclui biblioteca Util
4      funcao inicio()
5      {
6          real lado,area
7          inteiro prog=1
8          enquanto (prog == 1)
9          {
10             escreva ("Digite a medida do lado de um quadrado.\n")
11             leia (lado)
12             area = lado*lado
13             limpa()
14             escreva ("A área do quadrado é ",area,".\n")
15             Util.aguarde(1500) // Aguarda 1500 milisegundos (1,5 segundos)
16             escreva ("Deseja continuar executando este cálculo ?\n")
17             escreva ("Para continuar digite 1 ou 0 para encerrar.\n")
18             leia (prog)
19             limpa()
20         }
21     }
22 }
```

Exemplos

- **Limite de Armstrong**

1. O Limite de Armstrong consiste em uma situação combinada de altitude e pressão que faz com que a temperatura de ebulição da água caia para 37°C , igualando-se à temperatura interna do corpo humano. Assim, uma pessoa sem um traje pressurizado não sobreviveria em tais condições, pois seus fluidos corporais simplesmente iriam ferver.
2. Esse limite de altitude é alcançado quando a pressão atmosférica é tão baixa que a água ferve à mesma temperatura do corpo humano (37°C).
3. A água ferve a 37°C quando a pressão atmosférica está em torno de um dezesseis avos da pressão ao nível do mar. Isso ocorre em uma altitude por volta de 18.000 metros acima do nível do mar.



Exemplos

1. Escrever um algoritmo que simule o limite de Armstrong

```
1  programa
2  {
3      inclua biblioteca Util
4      funcao inicio()
5      {
6          inteiro altitude=0
7          enquanto (altitude <= 18000)
8          {
9              escreva ("Analisando a altitude do foguete...",altitude," metros...\n")
10             se (altitude >=0 e altitude<=10000)
11             {
12                 escreva ("Condição do astronauta estável !!!\n")
13                 Util.aguarde(500) // Aguarda 500 milisegundos (0,5 segundos)
14             }
15             se (altitude >=10001 e altitude<=15000)
16             {
17                 escreva ("Condição do astronauta grave !!!\n")
18                 Util.aguarde(500) // Aguarda 500 milisegundos (0,5 segundos)
19             }
20             se (altitude >=15001 e altitude<=17999)
21             {
22                 escreva ("Condição do astronauta crítica !!!\n")
23                 Util.aguarde(500) // Aguarda 500 milisegundos (0,5 segundos)
24             }
25             altitude = altitude + 650
26             limpa()
27         }
28         escreva ("O astronauta morreu !!!.\n")
29     }
30 }
```

Exercícios de repetição

❖ Exercício 78

1. Construir um algoritmo que calcule a média aritmética de vários valores inteiros positivos.
2. O final da leitura acontecerá quando for lido um valor **negativo**.

❖ Exercício 79

1. Escreva um algoritmo que calcule a média dos números digitados pelo usuário, se eles forem pares e múltiplos de 5.
2. Termine a leitura se o usuário **digitar zero** (0)

❖ Exercício 80

1. Escrever um algoritmo que leia um número não determinado de valores e calcule a média aritmética dos valores lidos, a quantidade de valores positivos, a quantidade de valores negativos e o percentual de valores negativos e positivos.
2. Mostre os resultados.

Exercícios de repetição

❖ Exercício 81

1. Digite uma quantidade não definida de números inteiros e escreva somente os números do intervalo de 25 a 75, inclusive. Digite 0 (zero) para encerrar.

❖ Exercício 82

1. Escreva um algoritmo para ler 2 valores. Se o segundo valor informado for **ZERO**, deve ser lido um novo valor, ou seja, para o segundo valor não pode ser aceito o valor zero.
2. Mostrar o resultado da divisão do primeiro valor lido pelo segundo valor lido.

❖ Exercício 83

1. Escreva um algoritmo para ler as notas da 1a. e 2a. avaliações de um aluno. Calcule e imprima a média (simples) desse aluno. Só devem ser aceitos valores válidos durante a leitura (0 a 10) para cada nota.
2. Acrescente uma mensagem 'NOVO CÁLCULO (S/N)? ' ao final do algoritmo. Se for respondido 'S' deve retornar e executar um novo cálculo, caso contrário deverá encerrar o algoritmo.

Exercícios de repetição

❖ Exercício 84

1. Cálculo da área do retângulo.
2. Utilize um comando de repetição que permita realizar este cálculo por “n” vezes. Escolha um critério de interrupção deste programa.
3. Conte quantas áreas de retângulos são maiores que 5000 e mostre este resultado.

❖ Exercício 85

1. Verifique se um número é par ou impar.
2. Utilize um comando de repetição que permita realizar este cálculo por n vezes. Escolha um critério de parada deste algoritmo.
3. Conte quantos números pares e impares foram digitados e mostre este resultado.
4. Antes de encerrar o programa, pergunte se o usuário deseja executar este programa novamente.

Exercícios de repetição

❖ Exercício 86

1. Foi feita uma pesquisa entre os habitantes de uma região.
2. Foram coletados os dados de idade, sexo (M/F) e salário.
3. Faça um algoritmo que informe:
 - a) a média de salário do grupo;
 - b) maior e menor idade do grupo;
 - c) quantidade de mulheres com salário até R\$100,00.

Encerre a entrada de dados quando for digitada uma **idade negativa**.

❖ Exercício 87

1. Chico tem 1,50 metro e cresce 2 centímetros por ano, enquanto Zé tem 1,10 metro e cresce 3 centímetros por ano.
2. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que Zé seja maior que Chico.

Exercícios de repetição

❖ Exercício 88

- Foi realizada uma pesquisa de algumas características físicas da população de uma certa região, a qual coletou os seguintes dados referentes a cada habitante para serem analisados:
 1. **sexo** (masculino e feminino)
 2. **cor dos olhos** (azuis, verdes ou castanhos)
 3. **cor dos cabelos** (louros, castanhos, pretos)
 4. **idade**
- Faça um algoritmo que determine e escreva:
 1. A maior idade dos habitantes
 2. A quantidade de indivíduos do sexo feminino cuja idade está entre 18 e 35 anos inclusive e que tenham olhos verdes e cabelos louros.
 3. O final do conjunto de habitantes é reconhecido pelo valor **-1** entrada como idade.

Exercícios de repetição

❖ Exercício 89

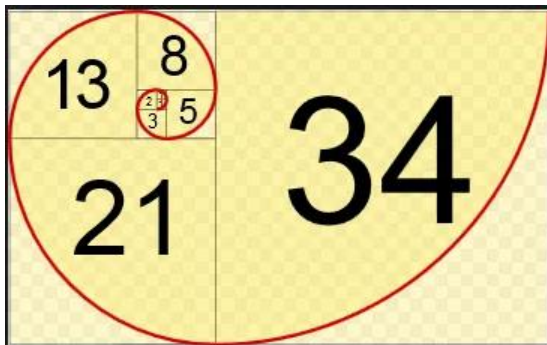
1. Em uma eleição presidencial existem quatro candidatos.
2. Os votos são informados através de códigos.
3. Os dados utilizados para a contagem dos votos obedecem à seguinte codificação:
 - 1,2,3,4 = voto para os respectivos candidatos;
 - 5 = voto nulo;
 - 6 = voto em branco;
4. Elabore um algoritmo que leia o código do candidato em um voto. Calcule e escreva:
 - total de votos para cada candidato;
 - total de votos nulos;
 - total de votos em branco;
5. Como finalizador do conjunto de votos, tem-se o valor 0.

Exercícios

Exercício 90

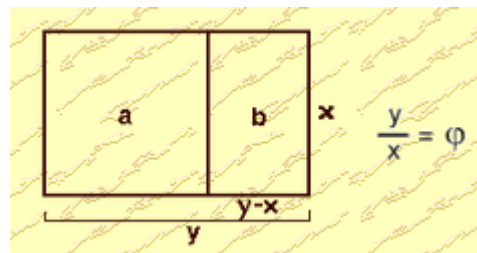
1. A série de Fibonacci é uma sequência de números naturais, na qual os primeiros dois termos são 0 e 1, e cada termo subsequente corresponde à soma dos dois precedentes.
2. Desenvolva um algoritmo que represente a série de Fibonacci até um número qualquer escolhido pelo usuário, que será o limite da série (fim).
3. Série ou sequencia de Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...



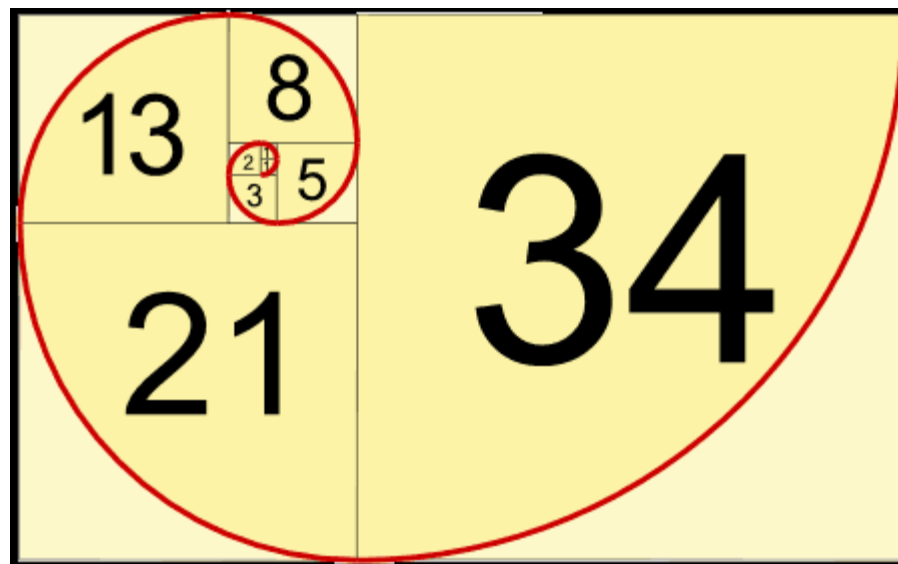
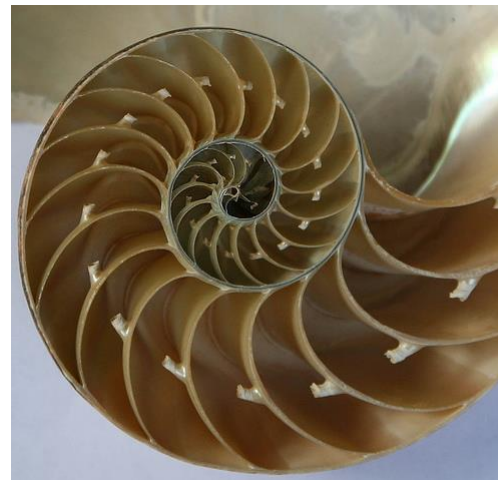
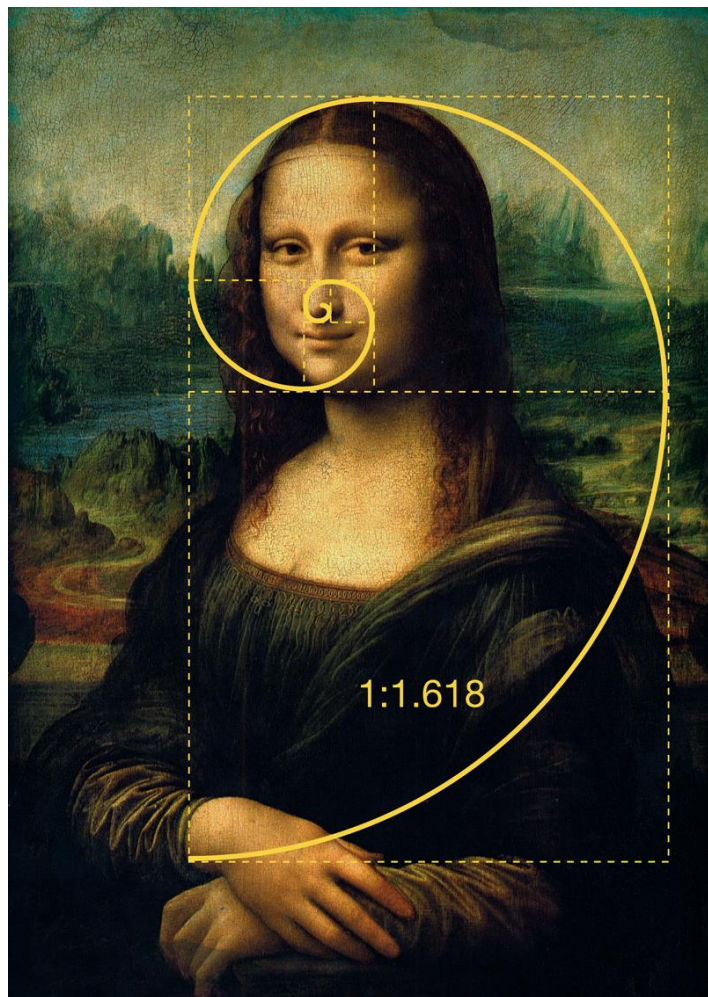
Exercícios

- **O Número de Ouro, Proporção Áurea e Fibonacci**
- Desde a Antiguidade, a proporção áurea é usada na arte. É frequente a sua utilização em pinturas renascentistas, como as do mestre Giotto.
- Este número está envolvido com a natureza do crescimento. Phi (não confundir com o número Pi), como é chamado o número de ouro, pode ser encontrado na proporção das conchas (o nautilus, por exemplo), dos seres humanos (o tamanho das falanges, ossos dos dedos, por exemplo) e nas colméias, entre inúmeros outros exemplos que envolvem a ordem do crescimento.
- Denomina-se **retângulo de ouro**, um retângulo que, quando é dividido em duas partes e em que uma dessas partes seja um quadrado, então o que resta terá que ser um retângulo com as mesmas proporções do retângulo inicial.



Exercícios

- Fibonacci e a proporção áurea





Obrigado !

Professor Ricardo Demattê
IFSP – Unidade Salto