

INTRO-CS-2- Introdução a Algoritmos

1. Algoritmos
2. Fluxograma
3. Portugol Studio
4. Palavras reservadas
5. Variáveis
6. Esqueleto de um programa

1. Algoritmos

Sequencia de passos que visão atingir um objetivo bem definido;

- **Pessoas:** Tem inteligência e habilidade racional, ou seja fazem perguntas para se esclarecer;
- **Computador:** Não tem senso próprio, deve receber instruções explícitas e precisas(Algoritmo).

Um algoritmo deve ter 3 qualidades:

1. Cada passo do algoritmo deve ser uma instrução que possa ser realizada;
2. A ordem dos passos deve ser precisamente determinada;
3. O algoritmo deve ter fim.

Conceitos:

- **Sequenciamento:** Estabelece um padrão de comportamento. As ações devem ser executadas linearmente, em sequência, uma após a outra;
- **Teste seletivo:** determina qual conjunto de ações deve ser seguido, dependendo do resultado da condição resultar em verdadeiro ou falso;
- **Repetição:** mesmo trecho é repetido várias vezes até que a condição de parada seja alcançada.

EXEMPLOS DE ALGORITIMOS:

Trocar uma lâmpada:

1. Coloque uma escada embaixo da lâmpada queimada;
2. Pegar uma lâmpada nova da mesma potência da queimada;
3. Suba na escada até alcançar a lâmpada queimada;
4. Gire a lâmpada queimada no sentido anti-horário até que ela solte;
5. Posicione a lâmpada nova no bocal e gire no sentido horário até sentir o aperto;
6. Desça da escada e acenda no interruptor;
7. FIM

Fazer um miojo:

1. Pegar uma panela;
2. Colocar água;

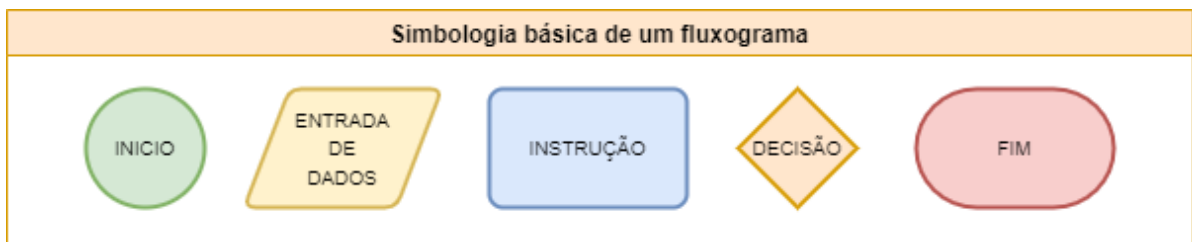
3. Acender o fogo;
4. Esperar a água ferver;
5. Colocar o miojo na água sem a embalagem;
6. Retirar do fogo após 3 minutos;
7. Colocar o tempero;
8. FIM

Trocar pneu de um carro:

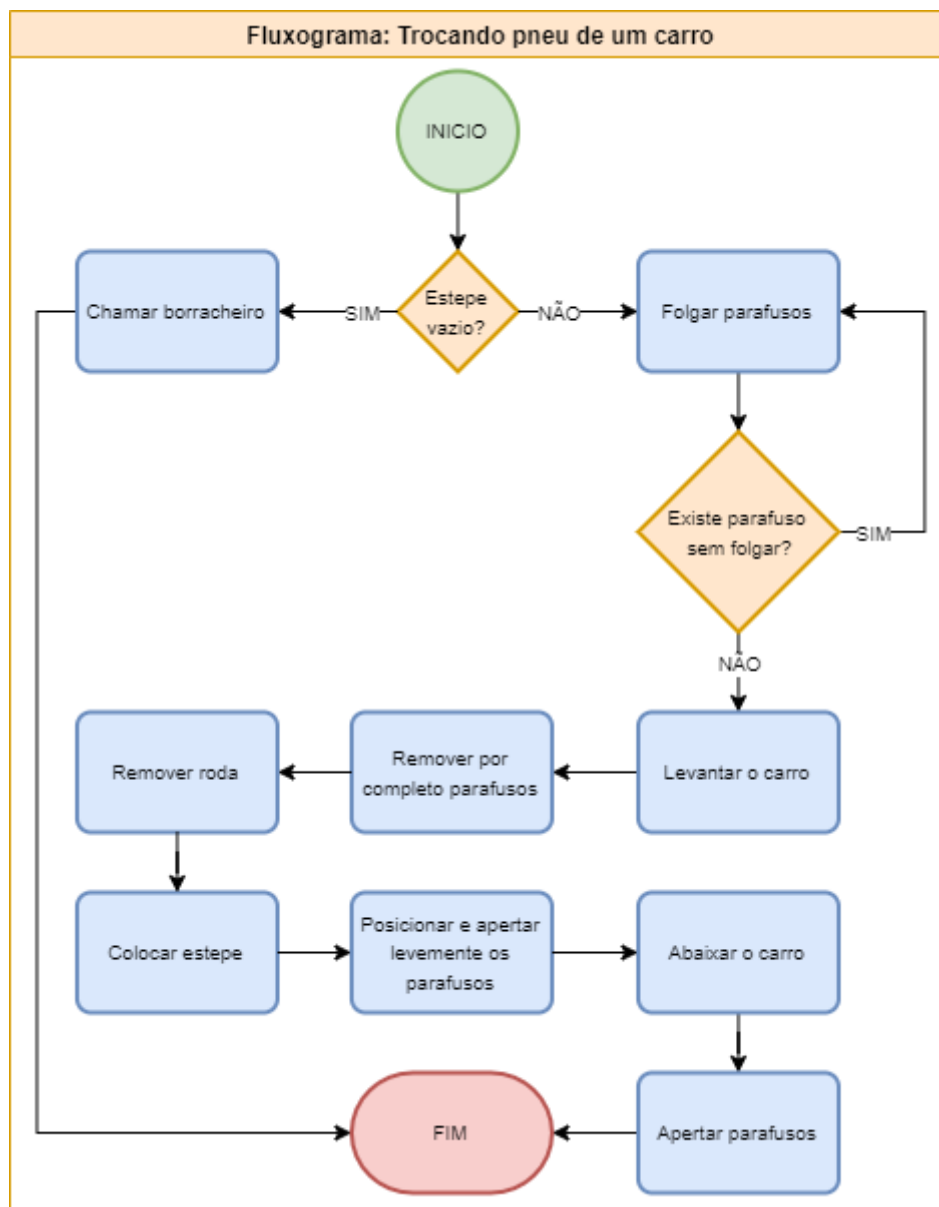
1. Validar se o estepe esta vazio;
2. Se não estiver, folgar parafusos do pneu furado;
3. Levantar o carro com o macaco;
4. Remover parafusos por completo;
5. Remover roda;
6. Colocar estepe;
7. Posicionar e apertar levemente parafusos;
8. Abaixar o carro;
9. Apertar novamente parafusos;
10. FIM.

2. Fluxograma

Representação esquemática de um algoritmo, feito através de gráficos que ilustram a transição de informações entre os elementos que o compõem. Abaixo segue simbologia básica para um fluxograma:



Para melhor compreensão segue um exemplo do algoritmo para troca de pneu representado em fluxograma:



3. Portugal Studio

Portugal é uma pseudolinguagem (linguagem de programação genérica e rica em detalhes) que permite ao leitor desenvolver algoritmos estruturados em português de forma simples e intuitiva. Para este curso estaremos utilizando o **Portugal Studio**. Esta ferramenta pode ser utilizada através do navegador pelo link [acessar aqui](#).

Saída de dados

Para fornecer uma saída de dados basta utilizar a palavra reservada **"escreva"**, veja o exemplo:

```
escreva("Primeira linha\nSegunda linha\n")
```

A palavra reservada **escreva** é utilizada para fornecer uma saída escrita no terminal do editor. É possível também fornecer variáveis como parâmetro, segue exemplo a seguir:

```
inteiro pontos = 10
escreva("Você fez ", pontos, " pontos!\n")
```

Basicamente, uma variável é uma informação que fica armazenada na memória. Pode ser um número, um texto, dentre outras coisas. No código acima, a variável `pontos` foi declarada e inicializada com o valor 10. Depois, usamos o comando `escreva()` para imprimi-lá 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. Variáveis serão cobertas em detalhes em outro tutorial. Por enquanto, você precisa apenas de uma noção básica desse conceito, que foi dada.

Entrada de dados

Para fornecer uma entrada de dados basta utilizar a palavra reservada **"leia"**, veja o exemplo:

```
inteiro x, y, idade
escreva("Digite a sua idade\n")
leia(idade)
escreva("Digite as suas coordenadas\n")
leia(x, y)
escreva("Coordenadas:\n", x, " e ", y, "\n", "Idade:\n", idade, "\n")
```

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.

O que o nosso algoritmo faz é bem simples:

- Solicita que o usuário digite uma idade.
- Lê a idade que o usuário digitou com o comando `leia()`.
- Solicita duas coordenadas e lê as duas de uma vez só com o comando `leia()` (é possível ler múltiplas variáveis de uma vez, separando as variáveis por vírgula).
- Usa o comando `escreva()` para imprimir o valor das variáveis.

4. Palavras reservadas

Palavras reservadas são componentes da própria linguagem e não podem ser redefinidas, ou seja, denominar elementos criados pelo programador. Toda linguagem possui palavras reservadas, cabe ao desenvolvedor efetuar um estudo sobre a linguagem para identificar as palavras reservadas. Para português segue uma lista das palavras reservadas pela linguagem para sua convenção:

Palavra	Descrição
pare	Faz o fluxo sair do bloco do código onde a palavra foi inserida.
caso	Executa um bloco de código dependendo da instrução switch.
continue	Faz com que o código existente após essa instrução não seja executado em um loop, indicando a próxima interação do loop.
caso contrario	Executará esse bloco de código se nenhuma das instruções switch/case for atendida.
faca	Executa um bloco de código um vez e, em seguida junto com a instrução while, processa um teste para determinar se o bloco deve ser executado novamente.
senao	Executa um bloco de código alternativo se uma condição se for falsa.
para	Usada para executar um loop condicional em um bloco de código.
se	Usada para executar um teste lógico verdadeiro ou falso.
return	Faz o fluxo sair de um método sem executar nenhum código que seja após a instrução (pode retornar uma variável opcionalmente).
escolha	Indica a variável a ser confrontada com as instruções caso.
enquanto	Executa um bloco de código repetidamente, enquanto um certa condição for verdadeira.

5. Variáveis

Variável é o nome dado a uma constante que recebe um tipo primitivo de dado. No portugol os tipos primitivos são:

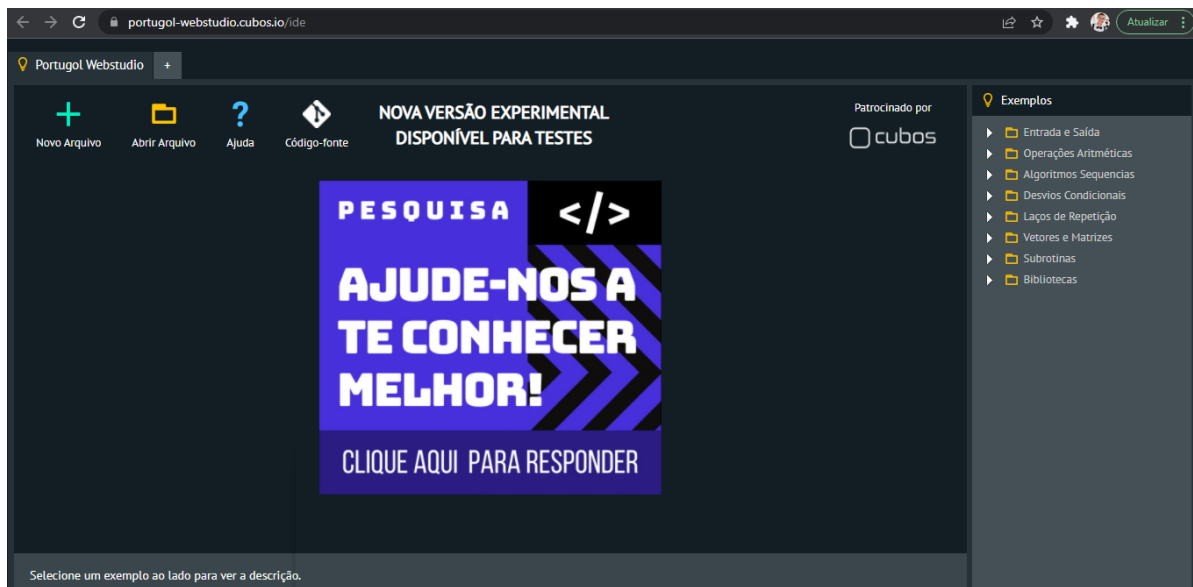
- Logico: Não é um valor numérico, só admite os valores verdadeiro ou falso;
- Caracter: Usa o código UNICODE e ocupa cada caractere 16 bits;
- Inteiros: Números inteiros, podem ser positivos ou negativos;
- Reais: números com casas decimais;

Uma variável em portugol é sempre declarada pertencente a um tipo. segue abaixo alguns exemplos:

```
inteiro x = 3 // declaração de uma variável 'x' iniciada a 3
real y = 2.5 // declaração de uma variável 'y', do tipo real iniciada em 2.5
logico lampadaAcesa = falso // declaração de uma variável 'lampadaAcesa', do tipo logico iniciada em falso
caracter c = 'κ' // declaração de uma variável 'c', do tipo caracter iniciada em 'κ'
```

6. Esqueleto de um programa

Acessando o Portugol Studio, [acessar aqui](#), é possível visualizar alguns esqueleto de programa na sessão exemplos:



No extremo direito é possível visualizar exemplos de código que é possível montar com Portugal Studio, navegue nos códigos para conhecer mais sobre a ferramenta. Disfrute e conheça a ferramenta!.

Você pode querer acessar um pequeno material para manipular masi o Portugal, segue o Link da [Documentação](#).