AULA 01

{introcomp}

ALGORITMOS E COMPUTAÇÃO

Algoritmos

- Estruturas Condicionais

Possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas.

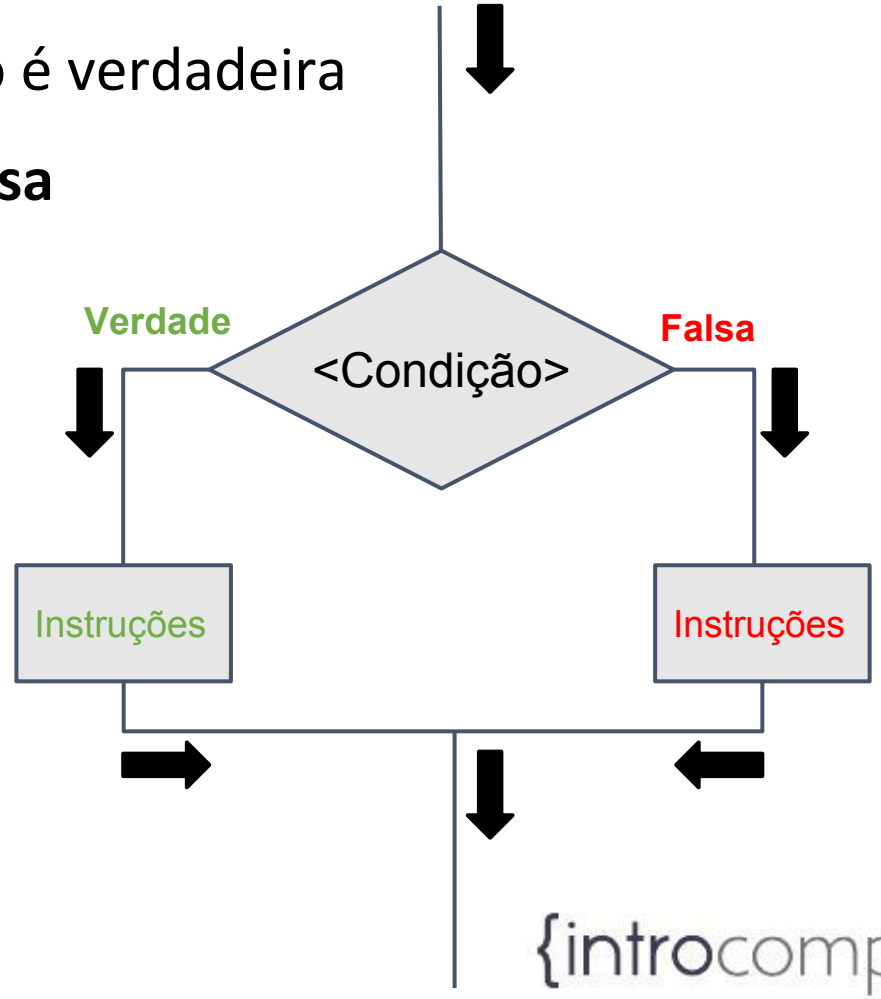
Por exemplo:

- Se não vier carro, continue. Caso contrário, permaneça parado;

Algoritmos

- Estruturas Condicionais
  - Acontecem apenas quando alguma condição é verdadeira
  - Uma condição só pode ser **verdadeira** ou **falsa**

```
se(condição) então
        <instruções>
senão
        <instruções>
fim-se
...
```



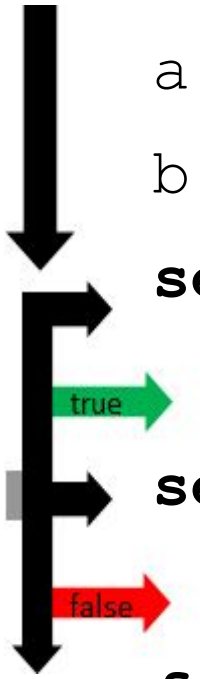
Algoritmos

- Estruturas Condicionais
  - Exemplo: Algoritmo que imprime qual o menor entre dois números

```
a <- leiaUmNumeroDoTeclado()
b <- leiaUmNumeroDoTeclado()
se(a < b)então
   menor <- a
senão
   menor <- b
fim-se
exiba menor
```

Algoritmos

- Estruturas Condicionais
  - Exemplo: Algoritmo que imprima qual o menor entre dois numeros




```
a <- leiaUmNumeroDoTeclado()
b <- leiaUmNumeroDoTeclado()
se(a < b)então
    menor <- a
senão
    menor <- b
fim-se
exiba menor
```

Hora de Praticar

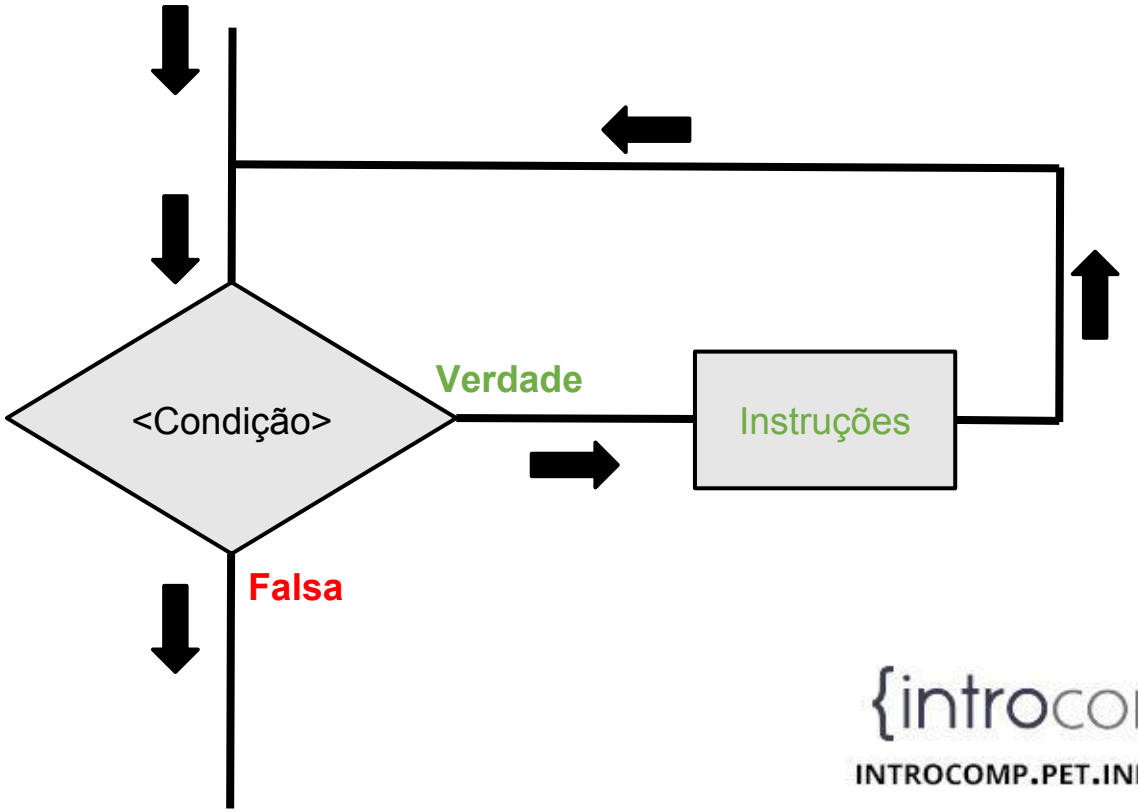
- **Exercício:**
  - Utilizando os conhecimentos que vocês acabaram de aprender façam no Runcodes:

  -> Praticando 1

Algoritmos

- Estruturas de Repetição
  - Irão repetir enquanto uma condição for satisfeita
  - São usadas as palavras chave **enquanto-faça**

```
enquanto (<condição) faça
   <instruções>
fim-enquanto
```



Algoritmos

- Estruturas de Repetição
  - Exemplo: Faça um algoritmo que imprima todos os números de 1 a 20

```
num <- 1
enquanto(num <= 20) faça
  exiba num
  num <- num+1
fim-enquanto
```

Algoritmos

- Estruturas de Repetição
  - Exemplo: Faça um algoritmo que imprima todos os números de 1 a 20


```
num <- 1
enquanto(num <= 20) faça
    exiba num
    num <- num+1
fim-enquanto
```

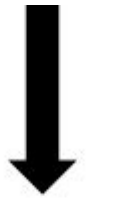
Algoritmos

- Estruturas de Repetição
  - Exemplo: Faça um algoritmo que some todos os números entre 200 e 400

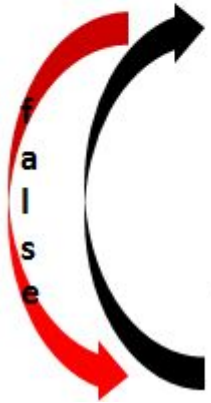
```
num <- 200
soma <- 0
enquanto(num <= 400) faça
    soma <- soma + num
    num <- num + 1
fim-enquanto
```

Algoritmos

- Estruturas de Repetição
  - Exemplo: Faça um algoritmo que some todos os números entre 200 e 400



```
num <- 200
soma <- 0
enquanto(num <= 400) faça
    soma <- soma + num
    num <- num + 1
fim-enquanto
```



Hora de Praticar

- **Exercício:**
  - Utilizando os conhecimentos que vocês acabaram de aprender façam no Runcodes:

  -> Praticando 2

Algoritmos

- Por que estudar algoritmos?
  - O seu impacto é amplo e de longo alcance.

Internet. Busca na web, roteamento de pacote, compartilhamento de arquivos,...

Biologia. Projeto do genoma humano, enovelamento de proteínas,...

Computador. Layout de circuitos, sistema de arquivos, **compiladores (fique ligado!)**...

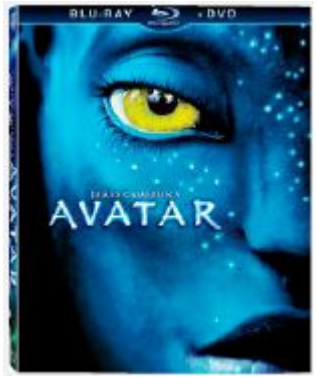
Computação gráfica. filmes, video games, realidade virtual...

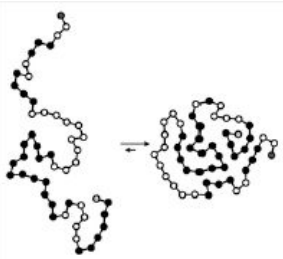
Segurança. celular, e-commerce, urna eletrônica,...

Multimídia. MP3, JPG, DivX, HDTV, reconhecimento de face, ...

Redes Sociais. recomendações, feeds de notícia, propagandas,...

Física. simulação de partículas, simulação de colisão de partículas,...





{introcomp}
INTROCOMP.PET.INF.UFES.BR

Linguagens de Programação

As pessoas se comunicam umas com as outras por meio de alguma linguagem. Podemos nos comunicar em português, inglês, francês, etc. Da mesma forma, precisamos de um jeito para nos comunicar com o computador. Para isso, utilizamos alguma linguagem de programação, como C, C++, Java, Python, PHP, Lua, etc.

Nesse curso, vamos aprender a linguagem C!

Compilador 

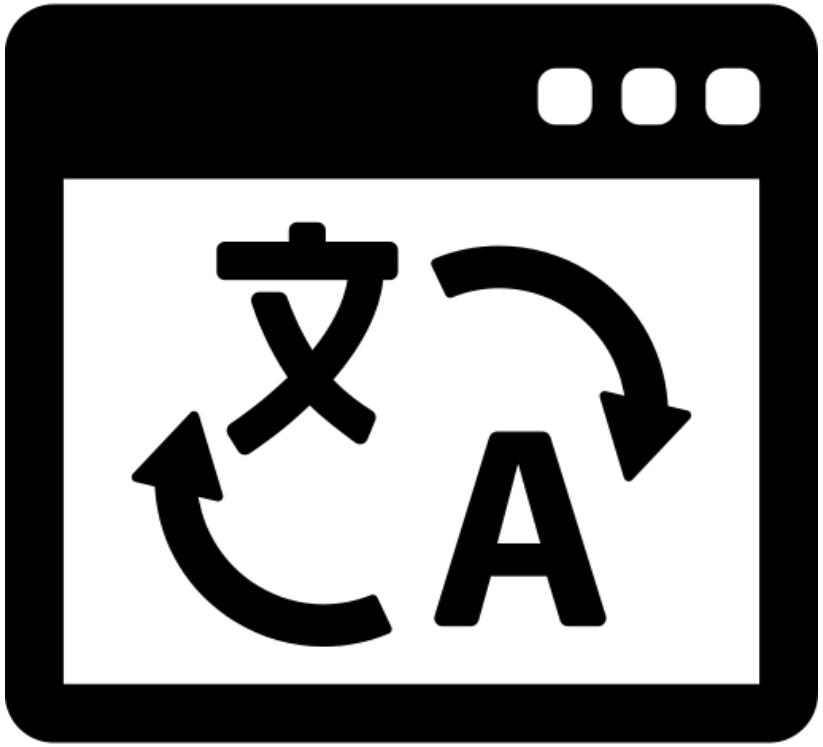
As linguagens citadas anteriormente são chamadas de linguagem de alto nível, pois usam palavras-chave muito próximas da realidade. Por exemplo: if/else, while, etc.

Compilador 

Entretanto, o computador precisa só entender linguagem de máquina. Dessa forma, precisamos de um “tradutor” que “traduza” nosso código em linguagem de alto nível para linguagem de máquina. O responsável por fazer isso é o compilador!

Linguagens de Programação

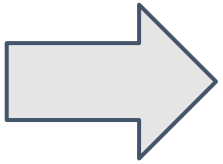
- Compilador
  - Traduz linguagem de alto nível (código fonte) em linguagem de de baixo nível (código objeto) na forma de um executável
  - Exemplos: gcc, Turbo C, Visual C, Visual Basic, etc
  - C é uma linguagem compilada!



Linguagens de Programação

```
#include <stdio.h>

int main()
{
    int a = 10, b = 5;
    printf (“%d\n”, a+b);
    return 0;
}
```

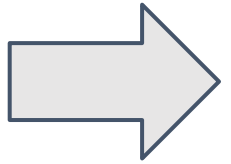


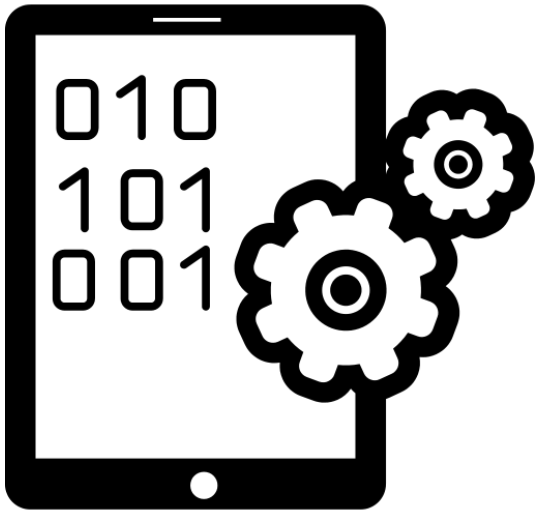
```
.data
x1: .word 10
x2: .word 5
x3: .word -1

.text
lw $t0, x1
lw $t1, x2
add $t2, $t0, $t1
sw $t2, x3

li $v0, 1
move $a0, $t2
syscall

li $v0, 10
syscall
```





AULA 01

{introcomp}

ALGORITMOS E COMPUTAÇÃO