

Período: 1º Semestre

Disciplina: Algoritmos e Lógica de Programação

Professor: Profa. Dra. Ligia Rodrigues Prete

E-mail: ligia.prete@fatec.sp.gov.br

2 – Introdução ao algoritmo

Você chegou até aqui porque
decidiu se tornar um
programador ou programadora.



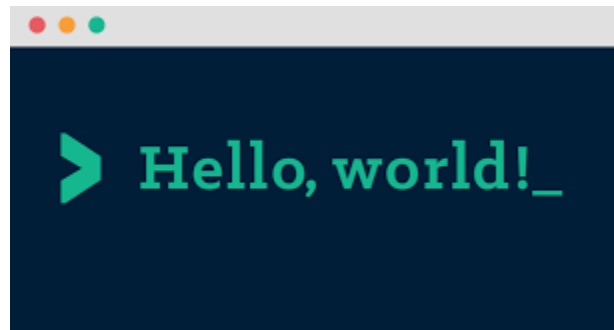


Então Parabéns!!

Essa é a profissão do futuro e que oferece muitas oportunidades de emprego com excelentes salários.

O que vamos aprender?

- Iremos aprender os primeiros conceitos necessários para entrar no mundo da programação.



Estes conceitos são:

1. O que é programar?
2. O que é ser um programador?
3. O que é um algoritmo?

1. O que é programar?

- Programar é realizar uma sequência de passos para desenvolver softwares, tais como ...

Sites



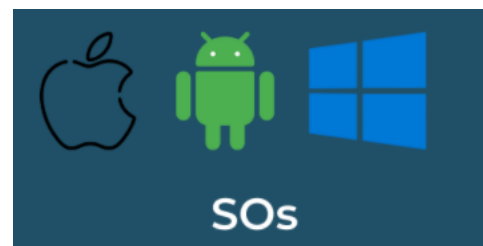
Aplicativos Móveis



Jogos



Entre outros ...



Por que programar é útil?

- O dispositivo (computador, celular, tablet, etc) não faz nada sozinho, por isso alguém precisa ensinar a ele os comandos para executar uma ação.

Exemplo 1

- A câmera de um celular tem a capacidade de tirar uma foto, mas precisa saber quando executar essa ação.



→ Tire a foto
quando o ícone
for tocado.

Exemplo 2

- Mesmo que um computador tenha um sistema operacional ele precisa saber quando iniciá-lo.

Quando o
computador ligar
inicie o sistema
operacional →



Como é feita a programação?

- A programação é feita a partir de blocos de código-fonte.

- Por exemplo, se quisermos imprimir na tela a mensagem “Olá mundo” criamos o seguinte código:

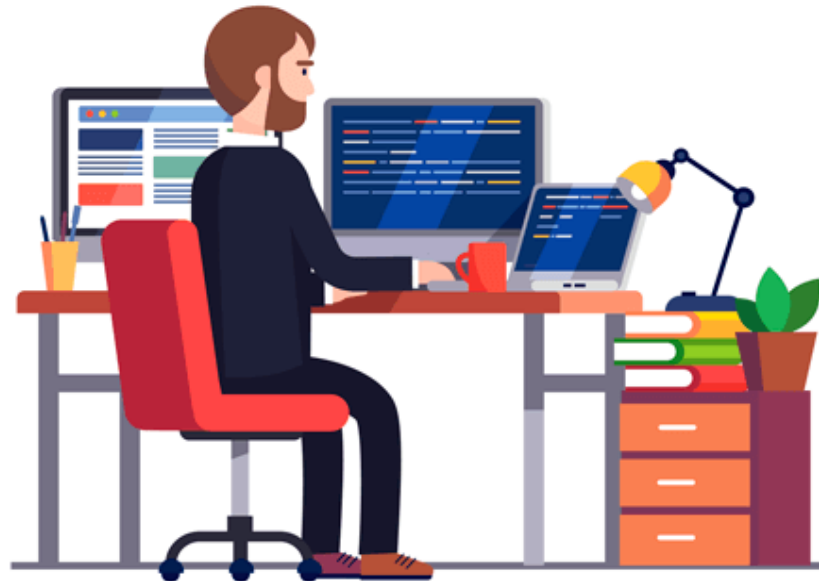
```
inicio()  
{  
    escreva(“Olá mundo!”);  
}
```

- Programar é o ato de escrever blocos de código-fonte.

```
string sInput;  
int iLength, iN;  
double dblTemp;  
bool again = true;  
  
while (again) {  
    iN = -1;  
    again = false;  
    getline(cin, sInput);  
    system("cls");  
    stringstream(sInput) >> dblTemp;  
    iLength = sInput.length();  
    if (iLength < 4) {  
        again = true;  
        continue;  
    } else if (sInput[iLength - 3] != '.') {  
        again = true;  
        continue;  
    } while (++iN < iLength) {  
        if (isdigit(sInput[iN])) {  
            continue;  
        } else if (iN == (iLength - 3)) {  
            continue;  
        }  
    }  
}
```


2. O que é ser um programador?

- É a pessoa que escreve blocos de código.
- Em outras palavras, é a pessoa que escreve os *softwares*.



Por que devo ser um programador?

- Tudo atualmente é feito através de um “dispositivo”, desde o pedido de uma pizza até o lançamento de um foguete.



- Por isso um programador tem muitas oportunidades de emprego com excelentes salários.



Onde vou atuar?

- Empresas públicas
- Empresas privadas
- Ser empreendedor

Plano de estudo para aprender a programar

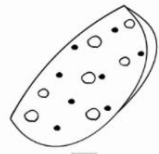
- Compreender o conceito de algoritmo.
- Treinar a lógica de programação.
- Aprender uma ou várias linguagens de programação.

3. O que é um Algoritmo?

- Algoritmo é uma sequência de passos determinados e organizados para executar uma determinada tarefa.

Exemplo 1

- Preparar um omelete:



Em uma cozinha
nosso objetivo é
preparar um prato



Um algoritmo é como
uma receita que
descreve os passos para
fazer esse prato (ou
resolver esse problema)



Se os passos na receita estão
corretos e foram seguidos o
resultado será um bom prato,
caso contrário teremos um
prato ruim ou diferente do
esperado.

Algoritmo para preparar um omelete.

Passo 1: Separe os ingredientes.

Passo 2: Quebre os ovos e misture com o queijo.

Passo 3: Coloque uma colher de manteiga na frigideira e aqueça.

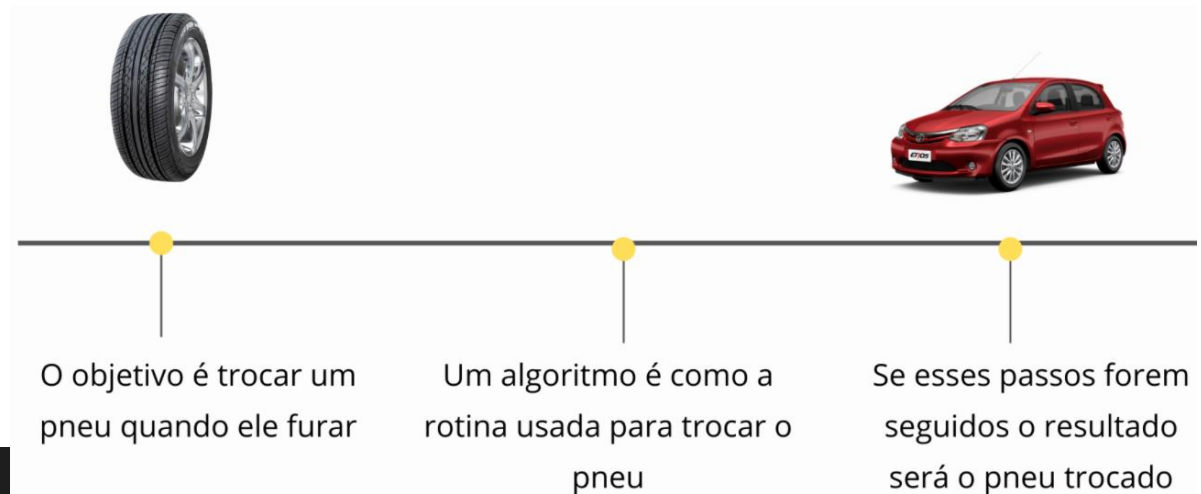
Passo 4: Coloque os ovos com o queijo na frigideira.

Passo 5: Deixe dourar dos dois lados.

Passo 6: Sirva.

Exemplo 2

- Trocar um pneu de carro:



Algoritmo para trocar o pneu

- Passo 1: Pegue o estepe, a chave de roda e o macaco hidráulico.
- Passo 2: Desaperte um pouco os parafusos do pneu.
- Passo 3: Erga o carro utilizando o macaco hidráulico.
- Passo 4: Termine de desapertar os parafusos.
- Passo 5: Substitua o pneu.
- Passo 6: Aperte um pouco os parafusos.
- Passo 7: Desça o carro.
- Passo 8: Termine de apertar os parafusos.

- Assim como as rotinas do nosso dia-a-dia um *software* também precisa de uma sequência de passos para seguir.

Exemplo 1

- Se precisarmos de um *software* que divida dois números e imprima o resultado na tela podemos descrever o seguinte algoritmo:

Algoritmo para dividir dois números

Passo 1: Peça para o usuário digitar o dividendo.

Passo 2: Guarde o dividendo.

Passo 3: Peça para o usuário digitar o divisor.

Passo 4: Guarde o divisor.

Passo 5: Faça a divisão entre dividendo e divisor e imprima o resultado na tela.

Exemplo 2

- Se precisarmos de um *software* para calcular a média de um aluno descrevemos o seguinte algoritmo:

Algoritmo para calcular a média do aluno

Passo 1: Peça para o usuário digitar a primeira nota.

Passo 2: Guarde a primeira nota.

Passo 3: Peça para o usuário digitar a segunda nota.

Passo 4: Guarde a segunda nota.

Passo 5: Calcule a média

$(\text{primeira nota} + \text{segunda nota}) / 2.$

Passo 6: Escreva o resultado na tela.

Algoritmo e Programação

- Primeiro passo é criar um algoritmo do *software*.
- Depois converte esse algoritmo para uma linguagem que o computador entenda (linguagem de programação).

Exemplo de um algoritmo aplicado na programação

Descrição Narrativa

Passo 1:

Peça para o usuário digitar o dividendo.

Passo 2:

Guarde o dividendo.

Passo 3:

Peça para o usuário digitar o divisor.

Passo 4:

Guarde o divisor.

Passo 5:

Faça a divisão entre dividendo e divisor

Passo 6:

imprima o resultado na tela.

Pseudocódigo

```
inicio()
{
    real dividendo
    real divisor

    escreva("Digite o dividendo")
    leia(dividendo);

    escreva("Digite o divisor")
    leia(divisor);

    escreva( dividendo / divisor )
}
```

Diagram illustrating the mapping of code lines to steps:

- escreva("Digite o dividendo") → Passo 1
- leia(dividendo); → Passo 2
- escreva("Digite o divisor") → Passo 3
- leia(divisor); → Passo 4
- escreva(dividendo / divisor) → Passo 5
- }

Praticando um Algoritmo

- Vamos criar um algoritmo para um *software* que imprime a idade do usuário.

- Vamos dividir o problema “**imprimir a idade do usuário**” em partes menores. Para isso iremos responder a seguinte pergunta ...

Qual é o problema?

Imprimir a idade do usuário.

Essa resposta vai gerar uma nova pergunta.

Qual é o problema?

Imprimir a idade do usuário.

Como sabemos a idade de alguém?

Diminuindo a data atual pela data de nascimento.

A medida que respondemos, novas perguntas vão surgindo.

Qual é o problema?

Imprimir a idade do usuário.

Como sabemos a idade de alguém?

Diminuindo a data atual pela data de nascimento.

Como descobrir a data atual?

Verificando a data do computador.

Como descobrir a data de nascimento de um usuário?

Pedindo a ele.

- Agora que temos as respostas podemos criar o nosso algoritmo, ou seja, definir o passo a passo até imprimir a **idade do usuário**.

Algoritmo descritivo para descobrir a idade de um usuário

1º passo: peça ao usuário para digitar a data de aniversário.

2º passo: guarde a data de aniversário.

3º passo: verifique a data atual do computador.

4º passo: guarde a data atual.

5º passo: subtraia a data atual pela data de aniversário.

6º passo: imprima a idade do usuário.

Dicas para criar um algoritmo

- Descubra qual o objetivo (problema).
- Comece a se questionar como resolver esse problema.
- Responda as perguntas que forem surgindo até não ter mais nenhuma.
- Baseado nas perguntas e respostas crie os passos (algoritmo) até chegar no objetivo ou resolver o problema.

Erros comuns

- Veja alguns erros na criação de um algoritmo para um software que imprime o valor final (com desconto) de um produto.

Não detalhar os passos

1º passo: peça ao usuário para digitar os valores.

2º passo: imprimir o valor final.

Como o *software* vai saber qual valor o usuário está digitando?

Pular uma etapa

1º passo: peça ao usuário para digitar o valor do produto.

2º passo: peça ao usuário para digitar a porcentagem de desconto.

3º passo: imprimir o valor final.

Mesmo com os valores (valor do produto e porcentagem de desconto) o *software* precisa calcular o desconto e o valor final antes de imprimir.

Ordem incoerente

- 1º passo:** peça ao usuário para digitar o valor do produto.
- 2º passo:** calcule o desconto.
- 3º passo:** peça ao usuário para digitar a porcentagem de desconto.
- 4º passo:** calcule o valor final.
- 5º passo:** imprimir o valor final.

Como o *software* vai calcular o desconto (passo 2) se ele ainda não sabe qual é a porcentagem (passo 3)?

Definir um passo incompleto

1º passo: peça ao usuário para digitar o valor do produto.

2º passo: peça ao usuário para digitar a porcentagem de desconto.

3º passo: calcule o desconto.

4º passo: calcule o valor final.

5º passo: imprimir o valor final.

O software precisa saber como calcular o desconto e o valor final.

Algoritmo descritivo correto

1º passo: peça ao usuário para digitar o valor do produto.

2º passo: peça ao usuário para digitar a porcentagem de desconto.

3º passo: calcule o valor do desconto:

$$(\text{porcentagem}/100) * \text{valor_produto}$$

4º passo: calcule o valor final:

$$\text{valor_produto} - \text{valor_desconto}$$

5º passo: imprimir o valor final.

Exercício 1

Algoritmo para sacar dinheiro se tiver saldo na conta:

Complete os passos com os códigos abaixo:

Passo 1: Vá até o caixa eletrônico

Passo 2: Insira o cartão.

Passo 3: Digite a senha.

Passo 4: Solicite o saldo

Passo 5: Verifique se o saldo é positivo

Passo 5.1: Se for solicite o saque e siga para o passo 6.

Passo 5.2: Senão encerre a operação e siga para o passo 7.

Passo 6: Retire o dinheiro do caixa eletrônico.

Passo 7: Retire o cartão.

Solicite o saldo

Verifique se o saldo é positivo

Vá até o caixa eletrônico

Exercício 2

Algoritmo para pendurar um quadro na parede.

Passo 1: Separar o parafuso, a furadeira, a bucha e a broca.

Passo 2: Colocar a broca na furadeira

Passo 3: Furar a parede que vai colocar o quadro

Passo 4: Apoiar a furadeira no chão.

Passo 5: Pegar a bucha e colocar no buraco feito.

Passo 6: Colocar o parafuso na bucha

Passo 7: Pendurar o quadro

Passo 8: Verificar se o quadro está firme.

Passo 8.1: Se estiver guardar a furadeira.

Passo 8.2: Senão colocar mais um parafuso.

Complete os passos com os códigos abaixo:

Furar a parede que vai colocar o quadro

Pendurar o quadro

Colocar a broca na furadeira

Colocar o parafuso na bucha

Exercício 3

Algoritmo para número par ou ímpar

Passo 1: Peça para o usuário digitar o número

Passo 2: Verifique se o usuário digitou o número.

Passo 2.1: Se digitou Guarde o número

Passo 2.2: Senão volte um passo.

Passo 3: Divida o número digitado por 2.

Passo 4: Guarde o resultado

Passo 5: Verifique se o resultado é inteiro.

Passo 5.1: Se for diga que o número é par.

Passo 5.2: Senão diga que é ímpar.

Complete os passos com os códigos abaixo:

Guarde o resultado

Guarde o número

Peça para o usuário digitar o número.

Exercício 4

Algoritmo para processo de votação

Passo 1: Eleitor vai até a urna

Passo 2: Eleitor digita o número do candidato

Passo 3: Eleitor confere as informações na tela

Passo 3.1: Se a informação estiver correta o eleitor confirma

Passo 3.2: Se a informação não estiver correta o eleitor aperta "Corrige" e volta ao passo 2

Passo 4: Com o voto confirmado o eleitor pode ir embora

Complete os passos com os códigos abaixo:

Se a informação estiver correta o eleitor confirma o voto

Eleitor digita o número do candidato

Com o voto confirmado o eleitor pode ir embora

Exercício 5

Algoritmo para trocar uma lâmpada

Passo 1: Desligar a luz/disjuntor

Passo 2: Pegar uma lâmpada nova

Passo 3: Pegar a escada

Passo 4: Subir na escada

Passo 5: Trocar a lâmpada

Passo 6: Descer da escada

Passo 7: Ligar a luz/disjuntor

Passo 8: Testar a lâmpada

Passo 9: Guardar a escada

Passo 10: Jogar a lâmpada antiga fora

Complete os passos com os códigos abaixo:

Subir na escada

Pegar uma lâmpada nova

Testar a lâmpada

Exercício 6

Algoritmo para verificar o número maior

Passo 1: Peça para o usuário digitar o primeiro número.

Passo 2: Guarde o primeiro número

Passo 3: Peça para o usuário digitar o segundo número.

Passo 4: Guarde o segundo número

Passo 5: Verifique se os números são iguais.

Passo 5.1: Se for imprima "os números são iguais".

Passo 5.2: Senão siga para o Passo 6.

Passo 6: Verifique se o primeiro número é maior que o segundo número.

Passo 6.1: Se for imprima "O primeiro número é maior que o segundo número"

Passo 6.2: Senão imprima "O segundo número é maior que o primeiro número".

Complete os passos com os códigos abaixo:

Se for imprima "O primeiro número é maior que o segundo número".

Guarde o primeiro número.

Senão imprima "O segundo número é maior que o primeiro número".

Exercício 7

Algoritmo para subtrair dois números

Passo 1: Peça para o usuário digitar um número

Passo 2: Guarde o primeiro número

Passo 3: Peça para o usuário digitar um segundo número

Passo 4: Guarde o segundo número

Passo 5: Faça a subtração

Passo 6: Imprima o resultado na tela

Complete os passos com os códigos abaixo:

Peça para o usuário digitar um segundo número

Imprima o resultado na tela

Guarde o segundo número