Lógica de programação

* Programar não é só ditar códigos, é resolver o problema. Não adianta criar o código, mas conseguir resolver o problema.
* O primeiro passo é aprender lógica e depois programar.

**O QUE É LÓGICA?**

* sequência corrente, regular e necessária de acontecimentos.
* Lógica de programação é contextualizar a lógica na programação d programadores. Você analisa a melhor sequência para resolver o problema.

**METACOGNIÇÃO**.

* Preciso comprar 2 camisetas. Elas custam 50 reais, totalizando 100 reais.
* Desconto é 20%. Quanto é isso? 20 reais.
* Há 2 maneiras de resolver o problema, uma mais completa e outra mais simples, removendo passos desnecessários, exemplo 80/100 = 100.08.

**ABSTRAÇÃO**.

* você coloca os aspectos essenciais de um contexto ignorando os aspectos menos importantes.
* Exercício: média aritmética de 4 notas: 5 +7 + 10 + 3 = 25/4 = 6,
* Fazer abstração: verificar se o método é o mais simples.

**ALGORÍTIMOS E PSEUDOCÓDIGOS**.

* algoritmo: sequencia de passos para resolver um problema.
* Ex: o que aconteceu no meu dia.

Inicio-dia

acordei

levante da cama

troquei de roupa

escovei os dentes

fui a padaria

tomei café

escovei os dentes novamente

fui ao trabalho

…

Fim-dia

* A ordem dos passos é importante e vai alterar o resultado.
* Site [studio.code.org/s/mc/stage/1/puzzle/1](http://studio.code.org/s/mc/stage/1/puzzle/1)
* Aqui consigo saber o que são os algoritmos

PSEUDOCÓDIGO

* uma forma genérica de escrever o algoritmo, utilizando uma linguagem simples nativa, de forma que qualquer pessoa consiga aprender.
* Apenas para aprender a programar.
* Os comandos na prática são em inglês.
* <https://www.proprofs.com/games/wolf-sheep-and-cabbage>
* Aqui, dá para escrever um pseudocódigo:

INICIO DO JOGO

ovelha entra no barco

atravessa

repolho entra no barco

atravessa

repolho sai do barco

atravessa

ovelha sai do barco

lobo entra no banco

atravessa

lobo sai do barco

atravessa

ovelha entra no barco

atravessa

ovelha sai do barco

Fim-Jogo

EXERCÍCIO FINAL:

* criar o algoritmo do dia:
* <https://studio.code.org/s/mc/stage/1/puzzle/1>
* Resolver os problemas em menos de 1 hora
* Fiz em cerca de 20 minutos

APRENDENDO FLUXOGRAMA, VARIÁVEIS E CONSTANTES.

* FLUXOGRAMAS
  + Representa graficamente o algoritmo. É o fluxo de dados.
  + Tem diferentes formas para diferentes tipos de processo.
  + Não existe um padrão para esse desenho.
* VARIÁVEIS
  + É um objeto, localizado na memória que retém, representa um valor ou expressão.
  + Ex: nota 1 recebe um valor (nota 7 por exemplo). Ex2: aluno: José da Silva.
  + Espaço da memória do computador que é alterado durante a execução do algoritmo.
  + Faz assim:

INICIO

DECLARA nota1: numero

DECLARA nota 2: numero

DECLARA media: numero

LEIA (nota1)

LEIA(nota2)

media=(nota1+nota2)/2

IMPRIMIR(media)

FIM

* se for assim:

INICIO

DECLARA pi=3,14

DECLARA raio= numero

FIM

* OBS: O PI será sempre o mesmo número. Por isso é constante. Já o raio é variável.
* <http://www.flowgorithm.org>
* Voce coloca uma

1. definição (começar sempre com uma letra e evitar acentuação) coloco as variáveis: nota1,nota2,media. Tipo de dado: real (números). (Essas são as variáveis declaradas). O usuário precisa digitar esses dados.
2. Entrada: forneço a variável que vai dar os dados= nota1
3. Crio duas entradas (nota1 e nota2)
4. Atribuir algo na variável media: media e a expressão será (nota1+nota2+total)/2
5. Saída: media

-

TOMADA DE DECISÕES E EXPRESSÕES

* são expressões que utilizam operadores aritméticos envolvendo constantes e variáveis

Ex: 50+50

total+50

* Soma +
* Subtração -
* Multiplicação \*
* Divisão /
* Potenciação ^
* Porcentagem %
* expressoes literais:

nome=“José da Silva”

Nome <- “José da Silva” (aqui define jose da silva para essa variável nome).

Media =(nota1+nota2)/2

PRINCIPAL

1. definição: a,b,c
2. Tipo: inteiros
3. Atribuição: a=2
4. Atribuição: b=a+3
5. Atribuição: c= a+b
6. Saida a
7. Saída b
8. Saída c

FIM

TOMADA DE DECISÃO

* 60% da programação
* Ex: quero desenvolver um site de vendas, somente para quem possui cartão de crédito. Se não tiver, tem que direcionar para outro site.

PRINCIPAL

caracteres cliente

lógico cartão

caracteres numero

entrada cliente

entrada cartão

alternativa (verdadeiro e falso) - cartão

FIM

COMO UTILIZAR A CONCATENAÇÃO

* unir uma sequencia de strings (sequência de caracteres).

PRINCIPAL

nome - caractere

saída - “qual é seu nome” (string)

entrada - nome

saída - “o seu nome é ” &(concatenação)nome