



Solução de problemas e pensamento computacional

DEFINIÇÃO

Apresentação do conceito de solução estruturada de problemas e pensamento computacional.

PROPÓSITO

Identificar os conceitos de solução estruturada e raciocínio lógico que permitem modelar problemas e obter solução computacional através da linguagem de programação.

PREPARAÇÃO

Antes de iniciar o conteúdo deste tema, tenha os softwares Bizagi e Portugol Studio instalados em seu computador.

OBJETIVOS

1º MÓDULO



- Identificar o conceito de solução estruturada, raciocínio lógico e pensamento computacional

2º MÓDULO



- Distinguir o uso de algumas ferramentas para a representação de soluções

3º MÓDULO



- Reconhecer as aplicações do pseudocódigo e da ferramenta Portugol Studio

MÓDULO 1

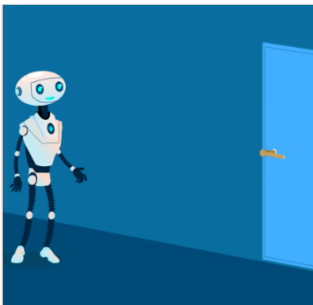
Identificar os conceitos de solução estruturada, raciocínio lógico e pensamento computacional

INTRODUÇÃO

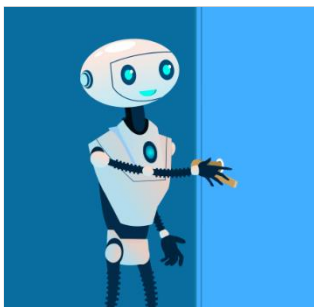
Caso você fosse questionado por um colega sobre como se abre uma porta, sua resposta seria algo parecido com “Segure a maçaneta, gire e puxe”. Logicamente, esta resposta é compreendida por quem perguntou, porque seu colega também é um ser humano!

Porém, ao tentar instruir um robô para executar esta tarefa, sua resposta não seria suficiente. O robô executa muito bem aquilo para o qual foi programado, mas não tem capacidade de compreender instruções vagas como “Segure a maçaneta, gire e puxe”.

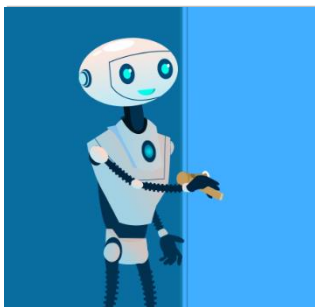
Para que a tarefa fosse corretamente desempenhada pelo robô, seriam necessários muitos mais níveis de detalhamento e definição, como uma sequência de passos. Poderíamos pensar em algo como:



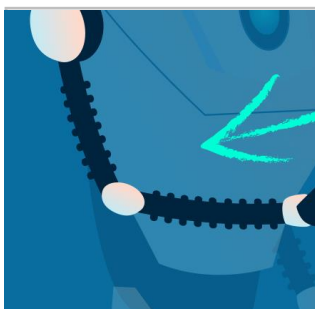
- Ande na direção da porta até que a distância fique igual a 30cm. Então, pare.



- Levante a mão direita até encostar em uma extremidade da maçaneta.
- Verifique se esta é a extremidade livre da maçaneta. Se for, passe para o próximo passo. Se não for, percorra a maçaneta até encontrar a extremidade livre.



- Feche os dedos envolvendo a extremidade livre da maçaneta.
- Exerça uma força de 10N sobre a extremidade livre da maçaneta para baixo até que ela não gire mais. Então, pare de puxar para baixo, mas continue segurando-a.



- Coloque seu cotovelo direito para trás sem soltar a maçaneta.

Observe como pode ser mais complexo dar instruções ao computador!

SOLUÇÃO ESTRUTURADA

O computador não tem discernimento para julgar culturalmente se algo é certo ou errado. No entanto, ele pode ser instruído para tomar decisões lógicas, baseadas em critérios objetivos, e seguir instruções ordenadas. A isso, chamamos **solução estruturada**.



Comentário

Embora possa parecer mais complexo dar instruções de forma estruturada a um computador, é só uma questão de prática para que você desenvolva esta habilidade. Além disso, existem diversas vantagens em instruir computadores a executar tarefas: eles não se cansam, não precisam parar para beber água, executam cálculos matemáticos muito mais rápido que seres humanos etc.

Propor uma solução estruturada para um problema consiste em elaborar uma sequência de passos a serem dados, de forma que, ao seguir esta sequência, exista uma resposta coerente para este problema.

Conforme foi dito, os computadores podem seguir instruções adequadamente programadas para realizar certas tarefas. Por isso, podemos concluir que eles têm comportamento previsível. Este fato nos permite *prever* o que determinado programa fará ao ser executado, caso tenhamos acesso ao conjunto de instruções que o gerou.



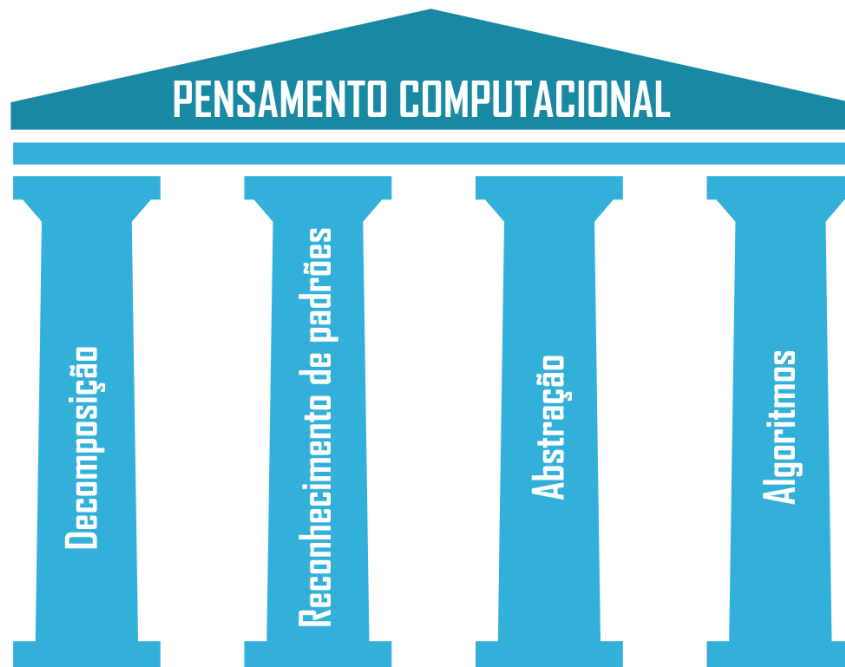
PENSAMENTO COMPUTACIONAL

Aplicar o **pensamento computacional** é o primeiro passo para conseguir instruir o computador a executar determinada tarefa. O pensamento computacional vai permitir que você proponha a solução de determinado problema de forma que o computador possa executá-la.

De acordo com Grover e Pea (2013), o pensamento computacional tem nove elementos que o levam a atender alunos interdisciplinarmente, além de avaliar seu desempenho. São eles:



De acordo com Brackmann (2017), o Pensamento Computacional tem quatro pilares para a resolução de problemas. São eles:



Decomposição: A decomposição consiste em dividir o problema inicial em partes menores, permitindo que cada parte menor seja mais facilmente resolvida. De maneira geral, problemas que não estão decompostos têm sua resolução mais difícil de enxergar e desenvolver.

Reconhecimento de padrões: Ao decompor o problema em partes menores, é possível aumentar a atenção aos detalhes e perceber que algumas destas partes menores já são conhecidas ou têm sua solução conhecida. O reconhecimento de padrões se caracteriza por identificar repetições ou regras de recorrência, aumentando a chance de conhecer algumas soluções para problemas similares.

Abstração: A abstração consiste na filtragem e classificação dos dados, concentrando a atenção no que realmente é importante (BRACKMANN, 2017). Ela pode ser vista como o principal dos pilares, porque será utilizada em diversos momentos. Ao concentrar-se nos dados realmente importantes, a abstração permite que decisões sejam tomadas com maior qualidade.

Algoritmos: Os algoritmos são procedimentos para resolver um problema com ações a serem executadas e a ordem em que elas devem acontecer. Na vida acadêmica, aprendemos diversos algoritmos no Ensino Fundamental e no Ensino Médio, como algoritmos para executar as operações básicas (soma, subtração, multiplicação e divisão). Os algoritmos serão mais explorados no próximo módulo.

Acesse o **material online** para assistir o vídeo e saber mais sobre “Os quatro pilares do pensamento computacional”.

RACIOCÍNIO LÓGICO

Ao longo da vida acadêmica de um aluno, ao passar pelo Ensino Fundamental e Ensino Médio, é comum que ele se veja tentando reconhecer padrões em boa parte do tempo. Aprende-se a resolução de exercícios de Ciências Exatas, por exemplo, a partir da repetição de alguns passos a problemas similares. Isto é, ao se deparar com um problema de determinado tipo, o aluno reconhece que este obstáculo, para o qual ele já tem uma solução, encaixa-se em determinado padrão.

Por exemplo, observe a sequência:



O próximo termo é  , já que esta sequência tem como regra listar os números pares.

A seguir, vamos treinar um pouco mais as suas habilidades para o reconhecimento de padrões e estimular seu raciocínio matemático.

1. Encontre o próximo termo da sequência: 1, 3, 5, 7, 9,

Resposta: Observe que a sequência é dos números ímpares. O próximo ímpar depois de 9 é o 11.

2. Encontre o próximo termo da sequência: 1, 6, 12, 19, 27,

Resposta: Repare que a diferença entre dois elementos sempre aumenta uma unidade a cada par. O primeiro par (1 e 6) tem diferença 5. O próximo par (6 e 12) tem diferença 6. Como o par (19 e 27) tem diferença 8, a próxima diferença será 9. Assim, $27 + 9 = 36$.

VERIFICANDO O APRENDIZADO

1. Tendo em vista o pensamento computacional, qual das opções abaixo não é um pilar para este conceito?

a) Algoritmos

b) Adaptabilidade

c) Decomposição

d) Abstração

Comentário

Parabéns! A alternativa **B** está correta.

Os pilares do pensamento computacional são: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos.

2. O reconhecimento de padrões está presente em nossas vidas desde a Educação Básica. Pode-se afirmar que ele consiste em:

a) Dividir o problema inicial em partes menores.

b) Filtragem e classificação dos dados.

c) Uma sucessão ordenada e finita de passos.

d) Identificar repetições ou regras de recorrência.

Comentário

Parabéns! A alternativa **D** está correta.

O reconhecimento de padrões se caracteriza por identificar repetições ou regras de recorrência, aumentando a chance de se conhecer alguma solução para problemas similares.

MÓDULO 2

Distinguir o uso de algumas ferramentas para a representação de soluções

INTRODUÇÃO

Neste módulo, você vai conhecer algumas ferramentas que podem ajudar a representar soluções, permitindo uma melhor organização do seu raciocínio e aplicação do pensamento computacional que está desenvolvendo. Como já foi abordado, os computadores são muito bons em executar tarefas. Se organizarmos nossas soluções em passos e condições lógicas, os computadores poderão ser instruídos a segui-los e encontrar respostas de forma rápida e precisa.

CONCEITOS

Basicamente, vamos aprender um pouco mais sobre algoritmos e fluxogramas. São dois conceitos importantes, que vão ajudar na nossa vida com os computadores.

Como já vimos antes, algoritmos são procedimentos para resolver um problema com as ações a serem executadas e a ordem em que elas devem acontecer. Um exemplo bem fácil de identificar é a receita para preparação de um macarrão instantâneo:



- ✓ Colocar 400 ml de água em uma panela;
- ✓ Aquecer esta água até que comece a ferver;
- ✓ Colocar o macarrão instantâneo na água fervente;
- ✓ Aguardar 3 minutos;
- ✓ Adicionar o tempero em pó;
- ✓ Retirar o macarrão instantâneo da panela;
- ✓ Colocá-lo em um prato.

Neste caso, fizemos uma sucessão de passos simples, sem nos preocupar com verificações que poderiam ser feitas para aumentar o nível de detalhamento.

Poderíamos, por exemplo, acrescentar a verificação se o gás está chegando adequadamente ao fogão. Poderíamos, também, verificar se o macarrão realmente está pronto após os 3 minutos. É possível fazer várias adaptações para aumentar o nível de detalhamento. Mas a ideia principal é criar uma sequência de passos que oriente a execução da tarefa.

Vamos observar outro exemplo simples sobre divisibilidade.



O critério para verificar se um número é par ou não é ser divisível por 2. Poderíamos pensar que ser divisível significa ter resto zero na divisão. Logo, um possível algoritmo para responder a um amigo se o número que ele falou é divisível por 2 ou não vem a seguir:

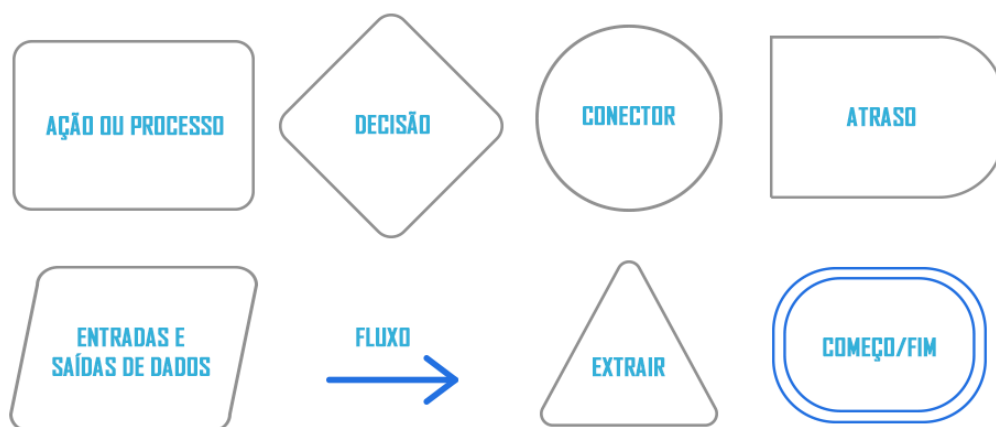
- Perguntar ao seu amigo o número que ele quer testar;
- Ouvir a resposta dele;
- Calcular o resto da divisão do número informado por 2;
- Se o resto da divisão do número informado por 2 for zero, então: **Responder ao amigo que o número é par;**
- Se não for: **Responder ao amigo que o número é ímpar.**

Um algoritmo que esteja organizado como o exemplo que acabamos de ver, já está bem próximo do que precisamos fazer para instruir corretamente um computador a executar esta tarefa.

FLUXOGRAMA

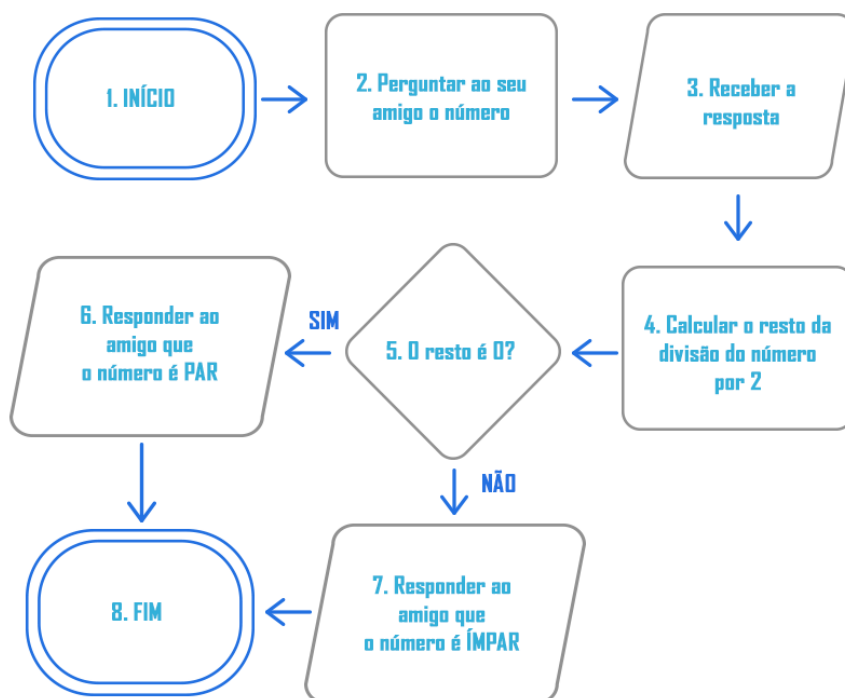
O fluxograma é outra ferramenta bem útil para a representação de uma solução, e pode ser entendido como a representação gráfica de um algoritmo. É como se pudéssemos fazer um esquema, com regras simples, que represente a sequência de passos e condições que compõem um algoritmo. Fluxogramas também podem ser usados para representar sistemas ou processos. Existem diversas ferramentas para a criação de fluxogramas, tanto locais como on-line. Podemos citar o Bizagi. Porém, até mesmo usando o Microsoft Word ou o Microsoft Powerpoint é possível criar fluxogramas.

Para utilizar corretamente o fluxograma, é necessário entender os seus principais componentes. São eles:

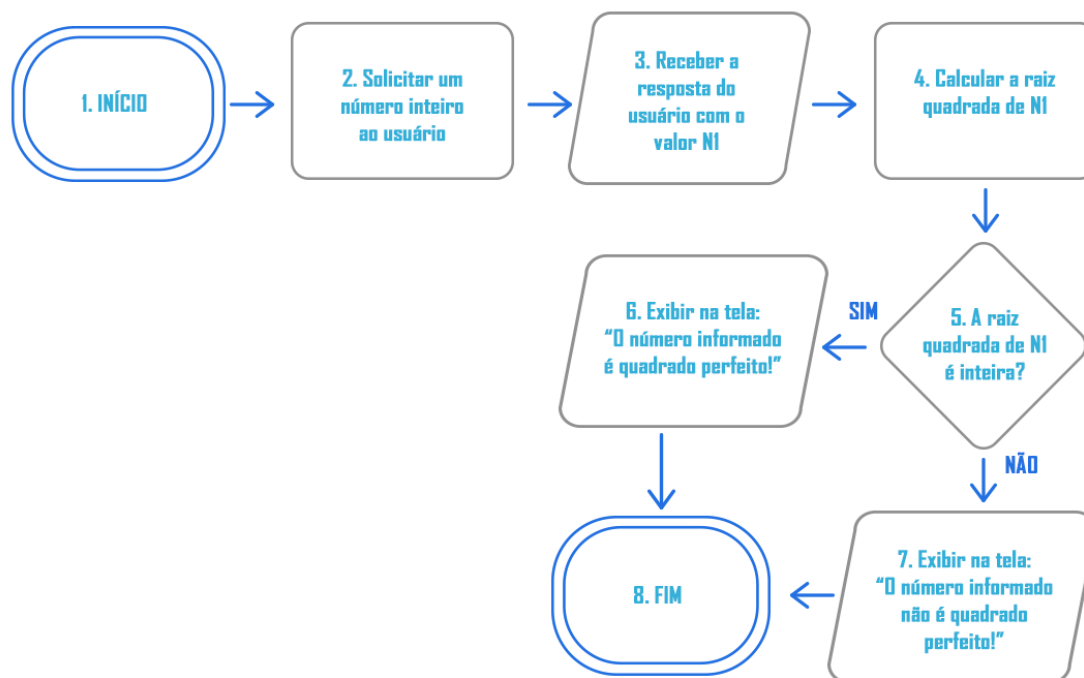


Acesse o **material online** para assistir o vídeo e **aprender a utilizar corretamente o Bizagi**.

O algoritmo que usamos para responder a um amigo, se o número que ele falou é divisível por 2 ou não, pode ser representado pelo seguinte fluxograma:



Considere o fluxograma a seguir:

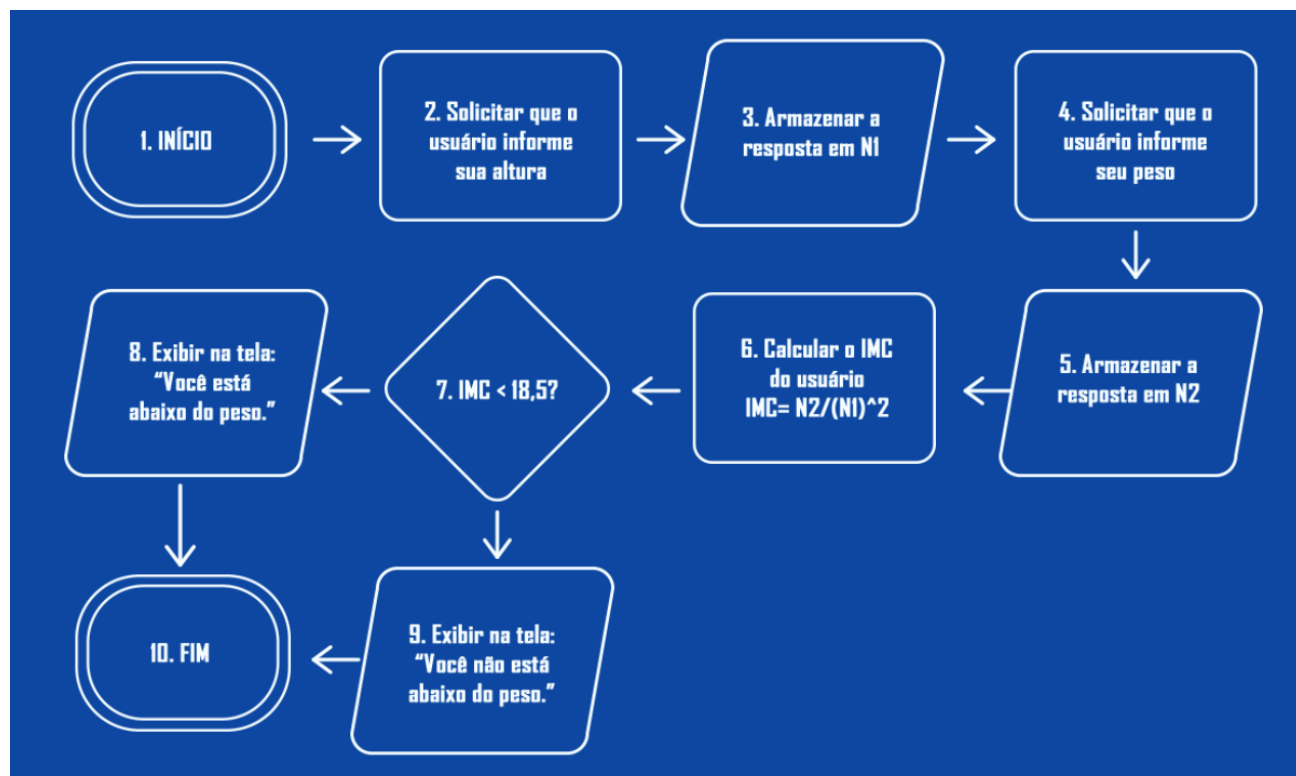


Supondo que o usuário informe, no passo 3, o número 125. Qual será o resultado deste fluxograma?

Resposta: O número não é um quadrado perfeito”, uma vez que 125 não tem raiz inteira. Assim, na decisão 5, o caminho será da seta não.

VERIFICANDO O APRENDIZADO

1. Considere o fluxograma a seguir:



Suponha que o usuário tenha informado o valor 1,80 no passo 3 e o valor 70 no passo 5. Qual será o resultado deste fluxograma?

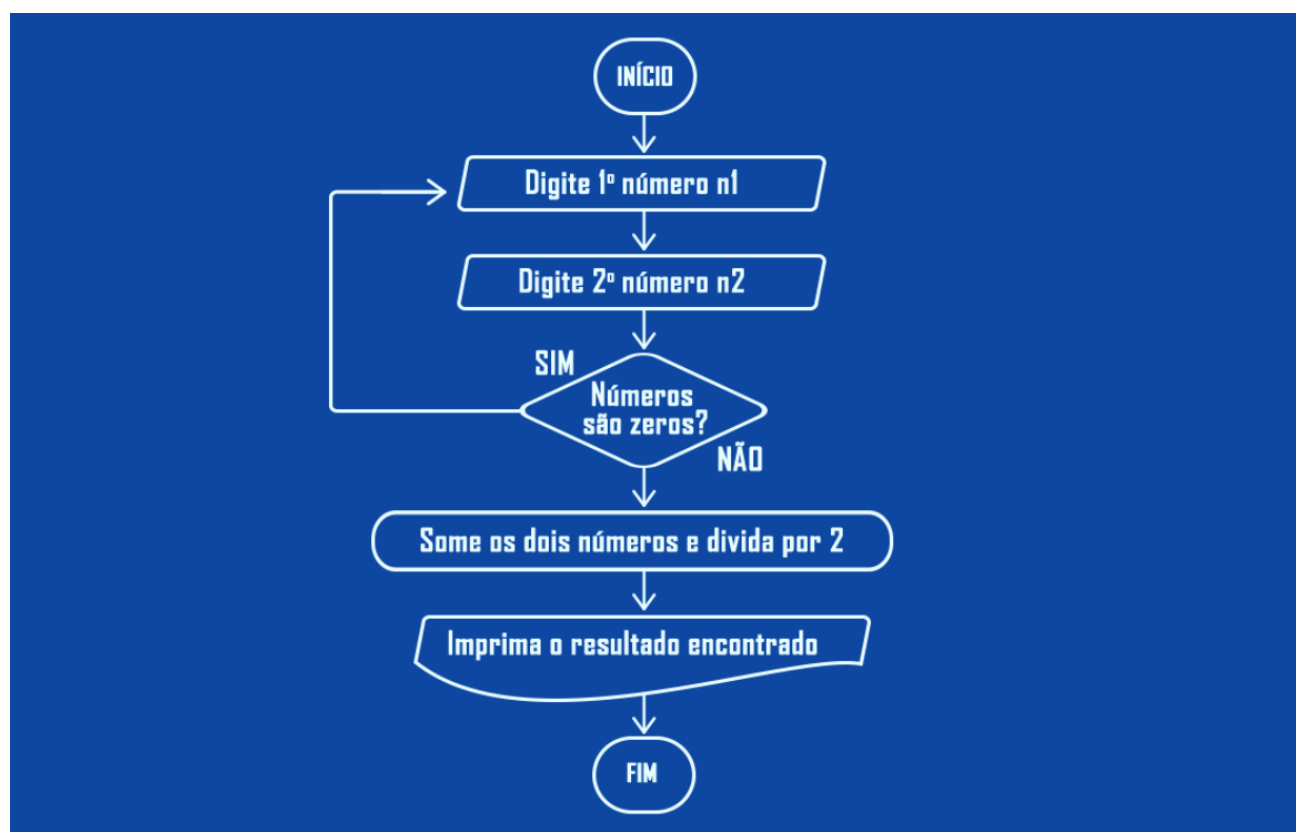
- a) Será exibido o valor 21,60.
- b) Será exibida a frase "Você não está abaixo do peso".
- c) Será exibida a frase "Você está abaixo do peso".
- d) Não é possível determinar o resultado apenas com estes dados.

Comentário

Parabéns! A alternativa **B** está correta.

Com os valores informados pelo usuário, o valor do IMC será 21,60. Porém, este valor não é exibido. Ele é utilizado na decisão 7, tomando o caminho da seta não. Assim, será exibida a frase na tela "Você não está abaixo do peso".

2. (FCC - 2016 - Adaptada) Considere o diagrama abaixo.



Analisando o raciocínio lógico e as estruturas lógicas utilizadas no diagrama, é correto afirmar que:

- a) A lógica implementa a solução de cálculo da média de 2 números diferentes de zero.
- b) Se um dos números digitados for zero, nada é impresso.
- c) Se os dois números digitados na primeira vez forem zero, os dois serão somados e divididos por 2.
- d) Se os dois números digitados forem iguais, nenhum resultado é impresso.

Comentário

Parabéns! A alternativa **A** está correta.

O fluxograma apresentado só terá algum resultado impresso caso os dois números digitados forem diferentes de zero. Neste caso, a impressão será do resultado da divisão da soma dos dois números por 2, que é justamente a média entre estes dois números.

MÓDULO 3

Reconhecer as aplicações do pseudocódigo e da ferramenta Portugol Studio

INTRODUÇÃO

Neste módulo, você conhecerá dois importantes aliados na sua iniciação à programação. Grosso modo, chamamos de *programar* a ação do programador ao, usando alguma linguagem de programação, criar um conjunto de instruções que um computador consiga executar. Tanto o pseudocódigo como o Portugol Studio permitirão que você se aproxime ainda mais da fluência na programação.

Pseudocódigo

Pseudocódigo é uma linguagem artificial e informal que ajuda programadores a desenvolver algoritmos. O pseudocódigo parece com a forma natural que usamos para nos expressar, mas também tem semelhanças com uma linguagem de programação para um computador real. Assim, ele é útil para ajudar o programador a pensar na solução, mas não pode ser executado por uma máquina.

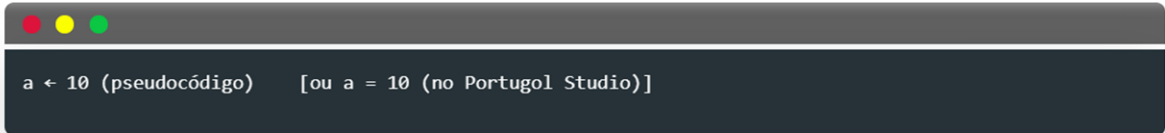
A seguir, um exemplo de trecho de programa em pseudocódigo:

```
Pedir um número inteiro ao usuário;  
Ler (a);  
Pedir um número inteiro ao usuário;  
Ler (b);  
Se (a >= b)  
    Imprimir a;  
Senão  
    Imprimir b.
```

Observação importante:

É muito comum utilizar o operador de atribuição, que em pseudocódigo, normalmente, é simbolizado pela seta (\leftarrow). No Portugol Studio, que você vai conhecer na sequência, o operador é simbolizado pelo igual (=).

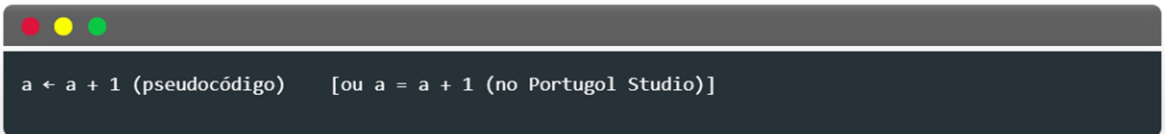
É importante perceber que o operador de atribuição não é uma igualdade. Ele atribui à variável do lado esquerdo o valor que está do lado direito. Ou seja, a linha:



```
a ← 10 (pseudocódigo)    [ou a = 10 (no Portugol Studio)]
```

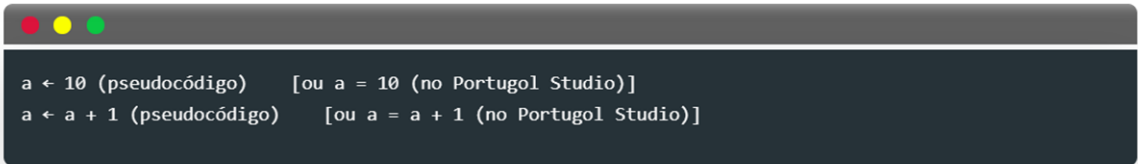
atribui o valor 10 à variável a.

É usual escrevermos a instrução



```
a ← a + 1 (pseudocódigo)    [ou a = a + 1 (no Portugol Studio)]
```

que causa estranheza no primeiro contato, mas simplesmente atribui à variável a o valor que ela tinha antes, acrescido de uma unidade. Ou seja, se as duas linhas acima estivessem em sequência, como a seguir:



```
a ← 10 (pseudocódigo)    [ou a = 10 (no Portugol Studio)]  
a ← a + 1 (pseudocódigo)    [ou a = a + 1 (no Portugol Studio)]
```

a variável a teria o valor 11 ao final da execução.

PORTUGOL STUDIO

O Portugol Studio é uma ferramenta para aprender programação, voltada para as pessoas que falam o idioma português. A ferramenta tem diversos exemplos e materiais de apoio à aprendizagem, o que facilita a jornada dos nossos futuros programadores. O Portugol Studio traz a funcionalidade de simular a execução real de um programa por meio de interações com o usuário, impressão na tela, entre outras.



Acesse o **material online** para assistir o vídeo e veja **um passo a passo para aprender a utilizar o programa**.

Exemplo de operações simples com números reais no Portugol Studio.

Explorando o exemplo de operações simples com números reais, que está na pasta Operações Aritméticas:

```
programa
{
    funcao inicio()
    {
        real a, b, soma, sub, mult, div

        escreva("Digite o primeiro número: ")
        leia(a)

        escreva("Digite o segundo número: ")
        leia(b)

        soma = a + b // Soma os dois valores
        sub = a - b // Subtrai os dois valores
        mult = a * b // Multiplica os dois valores
        div = a / b // Divide os dois valores

        escreva("\nA soma dos números é igual a: ", soma) // Exibe o resultado da soma
        escreva("\nA subtração dos números é igual a: ", sub) // Exibe o resultado da subtração
        escreva("\nA multiplicação dos números é igual a: ", mult) // Exibe o resultado da multiplicação
        escreva("\nA divisão dos números é igual a: ", div, "\n") // Exibe o resultado da divisão
    }
}
```

É possível observar o bloco chamado *programa*, que tem a função *inicio()*.

A função *inicio()* tem as variáveis reais *a*, *b*, *soma*, *sub*, *mult* e *div*.

A chamada *escreva* fará com que seja impressa na tela a frase que estiver entre aspas, dentro dos parênteses. Ou seja, *escreva("Digite o primeiro número: ")* fará com que seja impressa na tela a frase *Digite o primeiro número*:

A instrução

leia(a)

fará com que seja guardado na variável *a* o valor que for informado pelo usuário. Ou seja, após a sua execução, a variável *a* terá o valor que o usuário digitar.

A instrução

soma = a + b

faz com que a variável *soma* receba o conteúdo de *a + b*. Ou seja, os valores de *a* e *b* somados.

As 3 instruções seguintes são as atribuições dos valores correspondentes às variáveis *sub*, *mult* e *div*.

No final da linha, existe um texto explicativo, o qual chamamos de comentário, que será ignorado pelo Portugol Studio na hora de rodar o programa. Isto ocorre porque o texto está depois de duas barras (//), caracterizando o comentário. Ele serve para ajudar o programador a entender o código e organizar-se no futuro.

Observação importante: a variável *soma* tem esse nome, mas qualquer valor pode ser atribuído a ela. *Soma* é apenas um nome. É uma boa prática de programação usar nomes de variáveis que façam sentido. Ou seja, o nome escolhido é bom e ajuda a entender a finalidade de se ter esta variável. No entanto, poderia haver uma instrução:

soma = 0

O que atribuiria o valor 0 à variável *soma*. É importante que você entenda que o nome da variável ajuda a compreender o funcionamento do programa, mas o nome sozinho não garante que ela contenha o valor adequado.

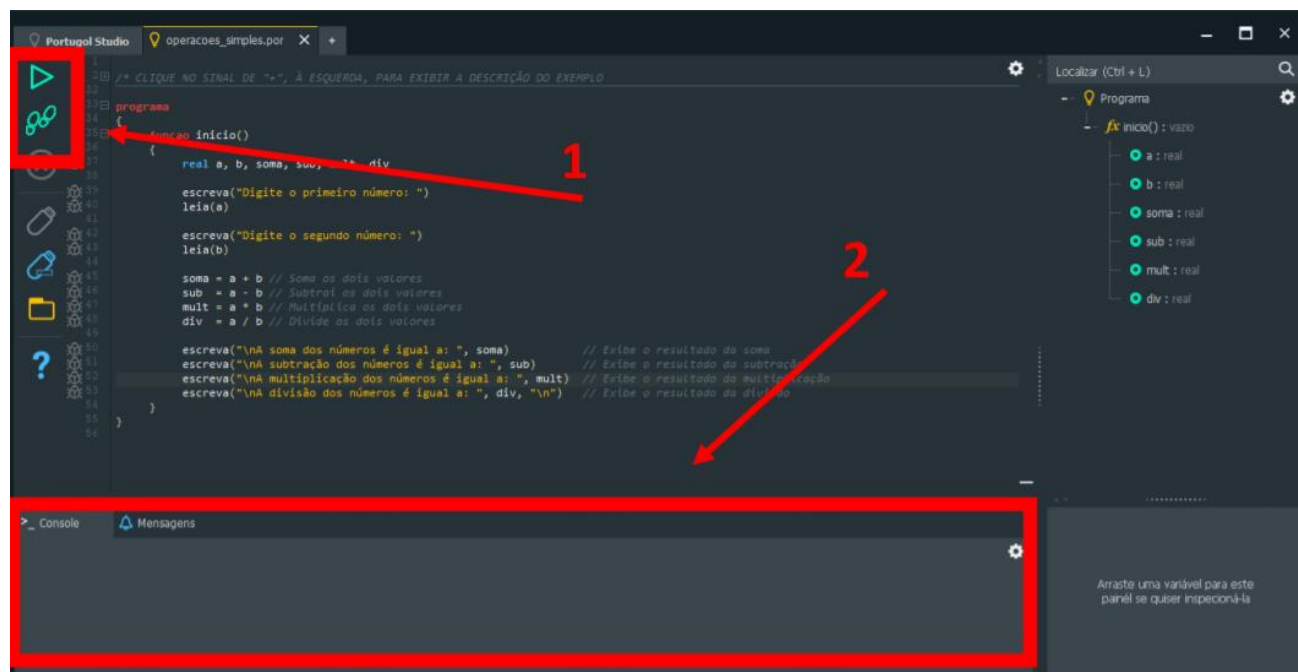
A instrução

```
escreva("\nA soma dos números é igual a: ", soma)
```

faz com que seja impressa na tela a frase

A soma dos números é igual a:

Mas observe que, depois das aspas, existe uma vírgula e uma variável (*soma*). Isto faz com que também seja impresso na tela, na sequência, o valor que está contido nesta variável. Além disso, dentro das aspas, existe o caractere especial `\n`, que faz com que o cursor passe para a próxima linha da tela.

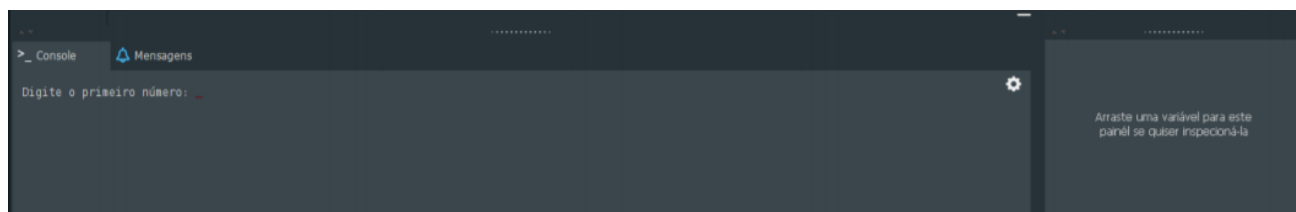


O retângulo 1 indica as duas formas de fazer o programa rodar: continuamente ou passo a passo. O retângulo 2 indica o console, que é a região em que serão exibidas as mensagens para o usuário e onde ficará o cursor para que ele digite o que for solicitado.

Ao executar a instrução

```
escreva("Digite o primeiro número: ")
```

é exibido no console o conteúdo que está entre aspas, como mostrado a seguir:



Recomendamos que o estudante tente se familiarizar com o Portugol Studio e veja os primeiros exemplos que já vêm com a ferramenta. Entender bem os conceitos iniciais e poder testar ajudará muito na sua formação como programador.

VERIFICANDO O APRENDIZADO

1. Observe o trecho de pseudocódigo a seguir:

$a \leftarrow 10$

$b \leftarrow 15$

$a \leftarrow a + b$

Imprimir (a)

Determine a saída de um programa que fosse escrito com este trecho.

a) 10

b) 15

c) 25

d) 1015

Comentário

Parabéns! A alternativa **C** está correta.

O comando de atribuição (\leftarrow) faz com que a variável **a** receba o valor da soma dos conteúdos das variáveis **a** e **b** neste momento (10 e 15, respectivamente). Ou seja, ao término desta atribuição, a variável **a** tem o valor 25.

2. Considere o seguinte programa, escrito em Portugol:

```
programa
{
    funcao inicio ()
    {
        inteiro numero

        escreva("digite um número inteiro: ")

        leia(numero)

        //TRECHO OMITIDO

    }
}
```

Na linha em que está escrito //TRECHO OMITIDO, deve ser inserida uma instrução que imprima na tela a seguinte frase: “O número digitado foi: , ” seguida do valor que o usuário digitou. Ou seja, se o usuário tiver digitado o valor 1 quando foi solicitado, a frase impressa na tela deve ser **O número digitado foi: 1** Para executar corretamente esta instrução, a linha omitida deve ser:

a) Escrever (“O número digitado foi: ”, numero)

b) Escreva (“O número digitado foi: ”, numero)

c) Escreva (“O número digitado foi: numero”

d) Escreva (“O número digitado foi: ”)

Comentário

Parabéns! A alternativa **B** está correta.

A função correta é *escreva()*, com os parâmetros passados entre os parênteses. O conteúdo da variável *numero* é o segundo parâmetro, sendo passado depois da vírgula, separado da frase estática “O número digitado foi:”.

CONSIDERAÇÕES FINAIS

CONSIDERAÇÕES FINAIS

Neste tema, você aprendeu como solução estruturada e raciocínio lógico embasam o que chamamos de pensamento computacional. A modelagem de soluções computacionais utilizando algoritmos e fluxogramas é altamente recomendada, antes que o programador efetivamente comece a desenvolver. Além disso, podem ser usados o pseudocódigo e o Portugol como passos intermediários.

O desafio desta disciplina consistiu em fazer você aprender a projetar e implementar soluções computacionais usando uma linguagem de programação. No mundo atual, cada vez mais vemos o crescimento do uso de dispositivos eletrônicos, como computadores, tablets e smartphones. Como um computador pode executar cálculos bilhões de vezes mais rápido que um ser humano, os profissionais que sabem programá-lo podem proporcionar enorme ganho de produtividade, o que os torna muito valorizados no mercado.

CONQUISTAS

- Identificou o conceito de solução estruturada, raciocínio lógico e pensamento computacional.
- Distinguiu o uso de algumas ferramentas para a representação de soluções.
- Reconheceu as aplicações do pseudocódigo e da ferramenta Portugol Studio.

REFERÊNCIAS

BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. Tese (Doutorado em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

GROVER, S.; PEA, R. **Computational Thinking in K-12: A Review of the State of the Field**. Educational Researcher, 2013, v. 42, n. 1, p. 38 - 43.

LIU, Yang. **Sistema de recomendação dos amigos na rede social on-line baseado em Máquinas de Vetores Suporte**. Dissertação (Mestrado em Informática) - Universidade de Brasília, Brasília, 2014.

MELVILLE, P.; SINDHWANI, V. ***Recommender systems***. *In*: Encyclopedia of machine learning. Berlin: Springer-Verlag, 2010.

EXPLORE +

Os exemplos do Portugol Studio, **Portugol Studio**. *In*: LITE da UNIVALI. Consultado em meio eletrônico em: 14 abr. 2020.

Acesse o tema online e imprima nossa **lista de exercícios complementares** para praticar o que aprendeu.