



Solução de problemas e pensamento computacional

Prof. Humberto Henriques de Arruda

Descrição

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, tenha os softwares Bizagi e Portugol Studio instalados em seu computador.

Objetivos

Módulo 1

Conceitos de solução estruturada

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

Módulo 2

Ferramentas de soluções

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

Módulo 3

Pseudocódigo e Portugal Studio

Reconhecer as aplicações do pseudocódigo e da ferramenta Portugal Studio.



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, essa 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 seja 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.

Observou como pode ser complexo dar instruções ao computador? É sobre a solução desse tipo de problema que vamos falar a partir de agora.

Bons estudos!



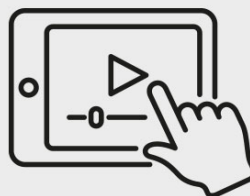
1 - Conceitos de solução estruturada

Ao final deste módulo, você será capaz de identificar o conceito de solução estruturada, raciocínio lógico e pensamento computacional.

Solução estruturada

Neste vídeo, vamos explorar a poderosa combinação entre a abordagem estruturada na programação e o raciocínio lógico. Aprenderemos como a solução estruturada nos auxilia a resolver problemas de maneira organizada e eficiente, enquanto o raciocínio lógico nos capacita a tomar decisões com base em condições e regras.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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 esse problema.

Já que os computadores podem seguir instruções adequadamente programadas para realizar certas tarefas, 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:

1. Abstração e reconhecimento de padrões;
2. Processamento sistemático da informação;
3. Noções de controle de fluxo em algoritmos;
4. Decomposição de problemas estruturados:

5. Sistema de símbolos e representações;
6. Pensamento iterativo, recursivo e paralelo:
7. Lógica condicional;
8. Eficiência e restrições de desempenho;
9. Depuração e detecção de erro sistemático.

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 se conhecer alguma solução para problemas similares.

Abstrações



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



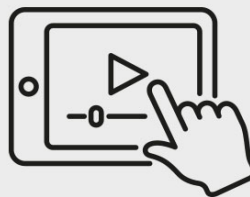
Os algoritmos são procedimentos para resolver um problema com as ações a serem executadas e a ordem em que elas devem acontecer. Na vida acadêmica, aprendemos diversos algoritmos no ensino fundamental e no ensino médio, como algoritmos para executar as operações básicas (soma, subtração, multiplicação e divisão).



Os quatro pilares do pensamento computacional

Assista ao vídeo e saiba mais sobre os quatro pilares do pensamento computacional.

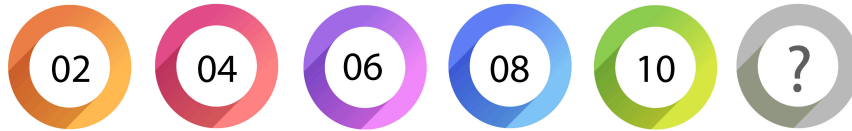
Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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**. Observe a sequência a seguir:



O próximo termo é (12), já que essa sequência tem como regra listar os números pares. Vamos agora treinar um pouco mais as suas habilidades para o reconhecimento de padrões e estimular seu raciocínio matemático. Vejamos:

Exemplo

Na sequência de números ímpares: 1, 3, 5, 7, 9,

O próximo numero seria: 11.

Na sequência de números: 1, 6, 12, 19, 27,

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, $27 + 9 = 36$.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Tendo em vista o pensamento computacional, qual das opções abaixo apresenta apenas pilares para este conceito?

A Algoritmos e adaptabilidade.

- B** Decomposição e eficiência.
- C** Reconhecimento de padrões e algoritmos.
- D** Algoritmos e eficiência.
- E** Abstração e adaptabilidade.

Parabéns! A alternativa C está correta.

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

Questão 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.
- E** identificar objetos e pessoas.

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.



2 - Ferramentas de soluções

Ao final deste módulo, você será capaz de distinguir o uso de algumas ferramentas para a representação de soluções.

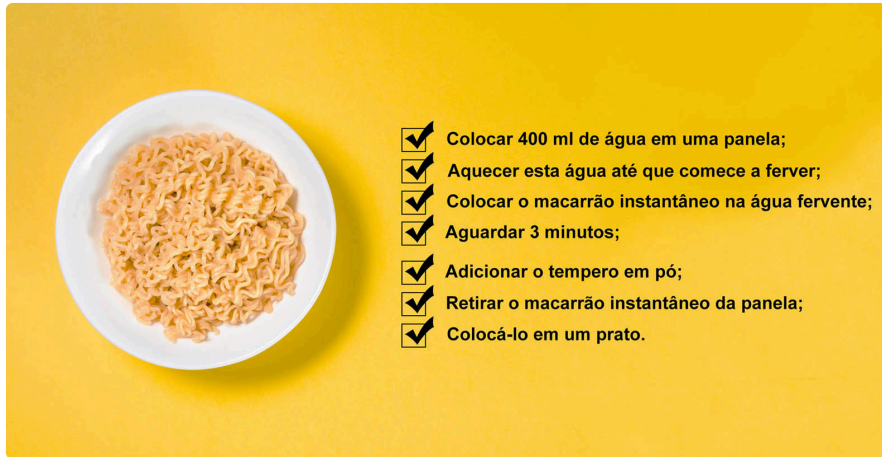
Conceitos

Conheça agora algumas ferramentas que podem ajudar a representar soluções, permitindo uma melhor organização do seu raciocínio e aplicação do pensamento computacional que você está desenvolvendo.

Sabemos que os computadores são muito bons em executar tarefas, e 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. Dessa forma, vamos aprender, neste módulo, um pouco mais sobre algoritmos e fluxogramas, dois conceitos

importantes que vão facilitar a nossa vida com os computadores.

Como já vimos, 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.



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.

Exemplo

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;

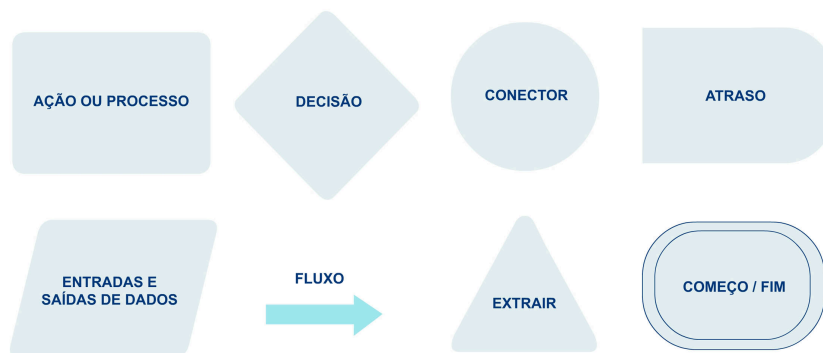
- Se não for: responder ao amigo que o número é ímpar.

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 a 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:



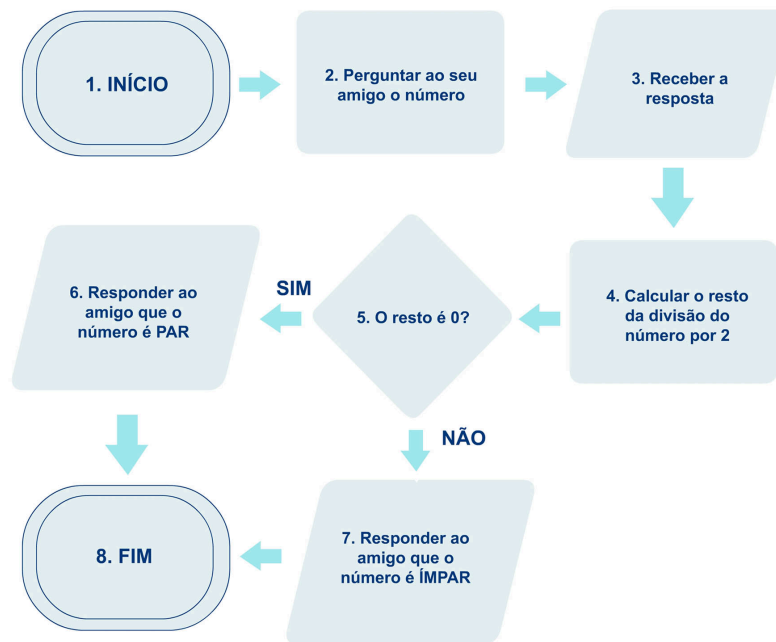
Como usar o Bizagi

Assista ao vídeo e aprenda a utilizar corretamente o Bizagi.

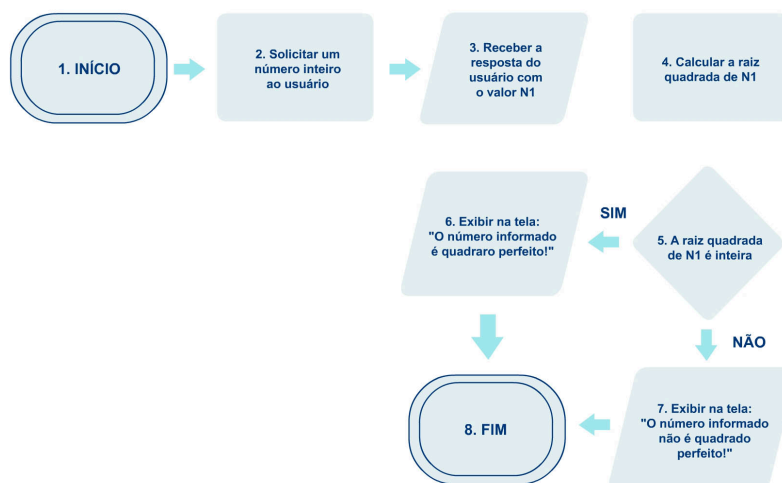
Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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:



Exemplo

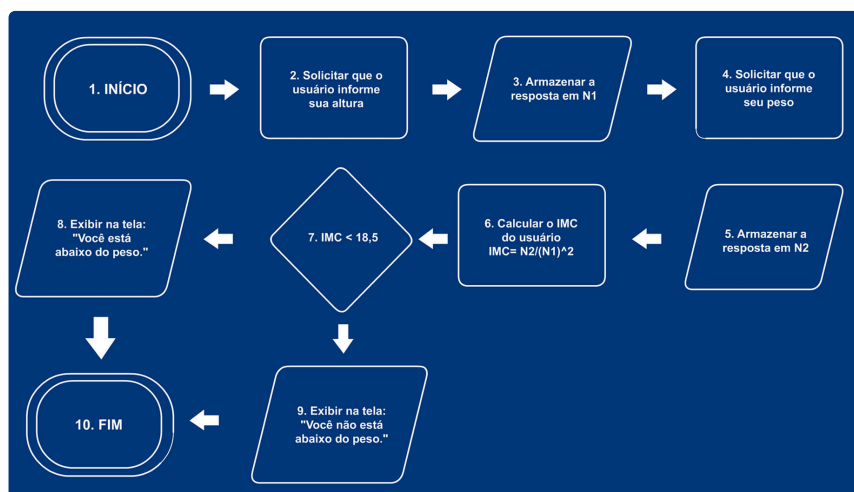
Supondo que o usuário informe, no passo 3, o número 125. Qual será o resultado deste fluxograma? O número não é um quadrado perfeito", uma vez que 125 não tem raiz inteira. Assim, na decisão 5, o caminho será da seta "não".

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 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".

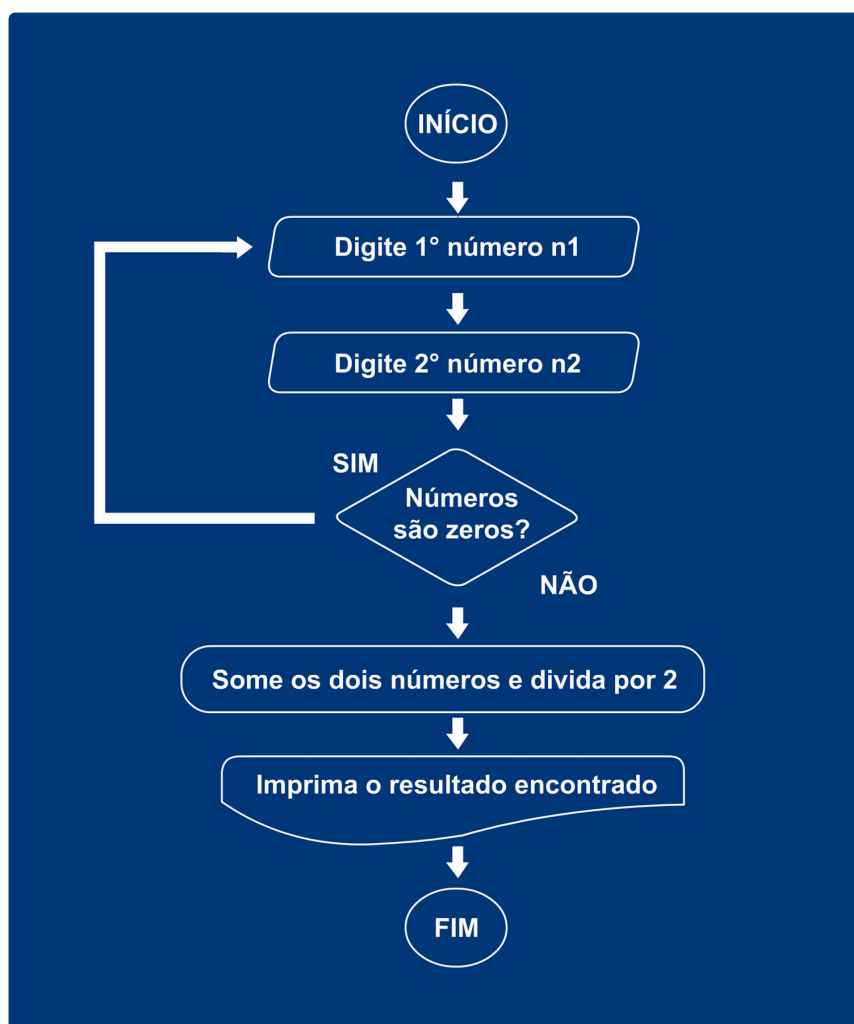
- D Será exibida o valor 18,5.
- E Não é possível determinar o resultado apenas com estes dados.

Parabéns! A alternativa B está correta.

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

Questão 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.
- E** a lógica apresenta o cálculo do MMC de um número.

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 esses dois números.



3 - Pseudocódigo e Portugol Studio

Ao final deste módulo, você será capaz de reconhecer as aplicações do pseudocódigo e da ferramenta Portugol Studio.

Pseudocódigo

Conheça agora dois importantes aliados na sua iniciação à programação. Grosso modo, chamamos de **programar** a ação do programador ao criar, usando alguma linguagem de programação, 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 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. Veja:

LINGUAGEM DE PROGRAMAÇÃO UTILIZADA



Observação:

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:

$a \leftarrow 10$ (pseudocódigo) [ou $a = 10$ (no Portugol Studio)]

atribui o valor 10 à variável a .

É usual escrevermos a instrução

$a \leftarrow a + 1$ (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:

$a \leftarrow 10$ (pseudocódigo) [ou $a = 10$ (no Portugol Studio)]

$a \leftarrow a + 1$ (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 sua jornada como futuro programador.



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.



Como usar o Portugol studio

Veja o passo a passo e aprenda a utilizar o programa.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Saiba mais

Consulte o [Exemplo de operações simples com números reais no Portugol Studio](#).

Procure 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.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 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

E 150

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.

Questão 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 essa instrução, a linha omitida deve ser:

- A escreva ('O numero digitado foi: 1', numero)
- B escreva (“O número digitado foi: ”, numero)
- C escreva (“O número digitado foi: numero”)

D escreva ("O número digitado foi: ")

E escreve ('O número digitado foi: ')

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

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.

Nosso desafio 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.



Podcast

Para encerrar, ouça um resumo dos principais tópicos abordados.

Para ouvir o *áudio*, acesse a versão online deste conteúdo.



Explore +

Confira o que separamos especialmente para você!

Veja os exemplos do **Portugol Studio** no site Lite da UNIVALI.

Acesse nossa [lista de exercícios complementares](#) para praticar o que aprendeu.

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.



Material para download

Clique no botão abaixo para fazer o download do conteúdo completo em formato PDF.



Download material

O que você achou do conteúdo?



Relatar problema