

L gica de Programac o

Aula 01 - Ol , Mundo! - Uma Introduc o Ao Mundo Da Programac o



Apresentação

Olá! Seja bem-vindo(a) à nossa primeira aula da disciplina Lógica de Programação! Nesta aula, você aprenderá os conceitos iniciais de Lógica de Programação, estudará a Linguagem de Programação, a maneira como construir o primeiro algoritmo, a ferramenta utilizada para sua correção e o modo como deve ser escrito o algoritmo na linguagem Potigol.

Neste momento, você já deve estar curioso(a) para saber o que é a Lógica de Programação, acertei? Ou, então, deve estar querendo saber para que ela serve, correto? Então, vamos lá!



Objetivos

Definir os conceitos relativos à Lógica de Programação.

Conhecer os conceitos referentes à Linguagem de Programação.

Identificar o funcionamento do computador como um interpretador.

Realizar a construção de um algoritmo e conhecer a ferramenta para execução dos exercícios de programação.

Executar o primeiro algoritmo com o Potigol.

Introdução à lógica de programação



“Ela, a **Lógica**, lhe dará a clareza de pensamento, a habilidade de ver seu caminho através de um quebra-cabeça, o hábito de arranjar suas ideias numa forma acessível e ordenada e, mais valioso que tudo, o poder de detectar **falácias** (É um argumento logicamente inconsistente, sem fundamento, inválido ou falho na tentativa de provar eficazmente o que alega.) e despedaçar os argumentos ilógicos e inconsistentes que você encontrará tão facilmente nos livros, nos jornais, na linguagem cotidiana e mesmo nos sermões e que tão facilmente enganam aqueles que nunca tiveram o trabalho de instruir-se nesta fascinante arte”

(Lewis Carroll)

O trecho que você acabou de ler é do famoso escritor Lewis Carroll, conhecido por utilizar conceitos da lógica e da antilógica para criar toda a fantasia do clássico “Alice no País das Maravilhas”. Antes de abordarmos especificamente a **Lógica de Programação**, que tal fazermos uma reflexão sobre o significado da palavra **Lógica**? O uso corriqueiro dessa palavra está normalmente relacionado à coerência e à racionalidade e frequentemente a associamos apenas à matemática. Você pode relacioná-la com a **"correção do pensamento"**, pois uma de suas preocupações é

determinar quais operações são válidas e quais não são. Uma vez que a forma mais complexa do pensamento é o raciocínio, podemos concluir que a lógica estuda a correção do raciocínio.

Existe lógica no dia a dia?

Acredito que, em certas ocasiões, a sua reação ao ler, ouvir ou ver alguma coisa foi falar a seguinte frase: **“Isso não tem lógica!”**, ou, então: **“Isso não é lógico!”** Em que você se baseou para fazer tais afirmações? Essas duas expressões lhe fazem pensar que a conclusão é indevida, portanto, o sentido deveria ser outro. É possível, também, que estejam indicando que o conhecimento sobre alguma coisa, sobre alguém ou sobre um fato não seja suficiente para você entender o que está ouvindo, vendo ou lendo. Ou seja, você utiliza a palavra lógica em seu dia a dia, para fazer afirmações e suposições de vários tipos e para tirar conclusões sobre determinados acontecimentos o tempo todo. Mas você pode perceber a importância da lógica em nossa vida não só na teoria, como também na prática, já que, para pensar, falar, escrever ou agir corretamente, precisa utilizá-la. Veja o exemplo abaixo:

Figura 01 - A lógica no cotidiano

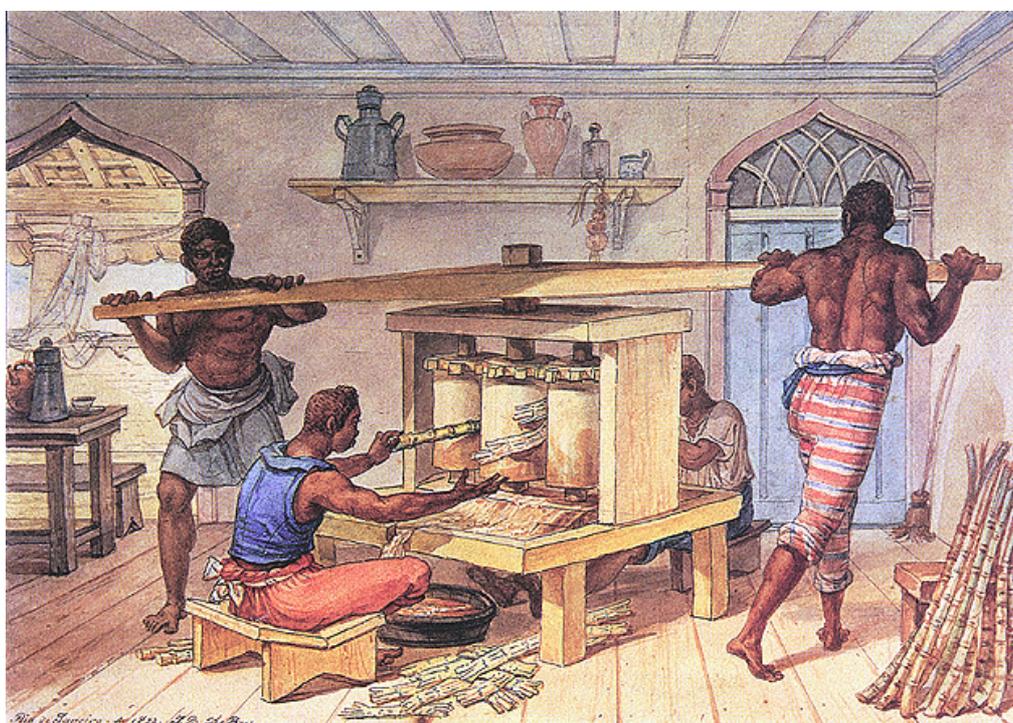


Percebeu que para a realização das ações feitas pelo personagem foi necessário seguir uma sequência lógica? Ou seja, a gaveta está fechada, mas a caneta está dentro dela, então foi necessário primeiro abrir a gaveta para depois pegar a caneta.

“Mas por que é importante saber **Lógica**, se o que eu quero é aprender a programar?” Não basta apenas saber programar, se você sempre fica confuso e enrolado nos códigos. Não basta aprender uma **Linguagem de Programação** e em seguida decorar todos os comandos. É necessário entender o que você pretende fazer. A lógica ajuda não só a programar, mas a encontrar erros, a pensar como você deverá enviar um comando, etc.

Vamos fazer um teste de lógica? Observe a imagem abaixo e descubra o erro. Mais especificamente, observe os dois homens que estão girando a [moenda](#) (A moenda é um tipo de mecanismo utilizado no engenho para moer ou espremer a cana de açúcar.) e o homem de blusa azul que está sentado colocando a cana de açúcar.

Figura 02 - Engenho manual que faz caldo de cana (Obra de Jean-Baptiste Debret)



Fonte: CRUZ, Pedro Oswaldo. Disponível em: . Acesso em: 04 out. 2017.

[Resposta ao teste](#)

O sentido de rotação por meio do qual os dois homens imprimem a moenda é incompatível com o sentido em que a cana-de-açúcar entra e sai dela. Eles fazem o cilindro central girar no sentido anti-horário. Desse modo, a cana deveria entrar pelo lado direito desse cilindro e sair pelo lado esquerdo. O que você observa é exatamente o contrário.

Agora, imagine que o pintor é um programador, cuja tarefa seja retratar o engenho manual de uma máquina de moer cana. Podemos dizer que ele atentou para a estrutura da máquina, mas não se ateu ao seu funcionamento. Ou seja, se você comparar a máquina a um programa, seria equivalente a estar com a sintaxe do código correta e mesmo assim o programa não funcionar, devido a sua sequência lógica.



Curiosidade

A descoberta desse erro lógico na aquarela do pintor francês Jean-Baptiste Debret foi feita por um aluno do curso de Engenharia Mecânica do Centro Federal de Educação Tecnológica de Minas Gerais, ao folhear um jornal que divulgava uma mostra do pintor. Para mais detalhes [acesse aqui](#).

E a lógica de programação?

Como o raciocínio é algo abstrato e intangível, os seres humanos se utilizam da capacidade de expressá-lo através da palavra falada ou escrita que, por sua vez, se baseia em um determinado idioma e segue uma determinada gramática. Assim, embora um mesmo raciocínio possa ser expresso em qualquer um dos inúmeros idiomas existentes, continuará representando a mesma ideia.

Algo similar ocorre com a **Lógica de Programação**, que pode ser concebida pela mente treinada e é capaz de ser representada em qualquer uma das inúmeras linguagens de programação existentes. Nesse sentido, o significado da Lógica de Programação diz respeito ao **uso correto das leis do pensamento e de processos de raciocínio e simbolização formais na programação de computadores, com vistas à racionalidade e ao desenvolvimento de técnicas que cooperem para a produção de soluções logicamente válidas e coerentes, capazes de resolver com qualidade os problemas a serem programados.**

Linguagens de programação existentes?

Você deve ter percebido que o título da aula começa com **Olá Mundo!** Por que esse título? É assim que o primeiro exercício básico da Linguagem de Programação é apresentado e aplicado. "Olá Mundo" (ou *Hello, World!* em Inglês) é um famoso programa de computador utilizado para testar a linguagem de programação e/ou um compilador. Mas antes de chegar a esse ponto, comece do princípio: os computadores sabem "falar"!

Isso mesmo! Assim como você sabe falar o idioma Português, os computadores "sabem falar" na língua deles. Para você se comunicar com os computadores, há diversos "idiomas" que podem ser utilizados, tais como a linguagem Java, a linguagem C ou C++, Ruby, Python e Assembly, dentre outros. Neste momento você deve estar se perguntando onde essas linguagens são utilizadas, certo? Comumente grandes empresas utilizam mais de uma linguagem em seus projetos. Por exemplo, a Google usa em sua ferramenta de busca JavaScript, C, C++, Java e Python. Já o Facebook trabalha com o PHP, Erlang, Java, Hack, C++. A maioria das empresas utiliza mais de uma linguagem para concretizar a construção de seus projetos.

Apesar disso, é importante ressaltar que os computadores são extremamente “burros” e que, de fato, eles não pensam sozinhos. Você deve estar pensando: “Poxa, além de ter de domesticá-los, ainda os chamamos de burros?” Calma, não é nada de *bullying* ao computador. É apenas uma forma de expressão que utilizamos ao lidar com as máquinas. Ou seja, quando você precisar dar uma instrução ao computador, essa instrução deve estar clara e bem definida, caso contrário a máquina não lhe dará a resposta que você espera.

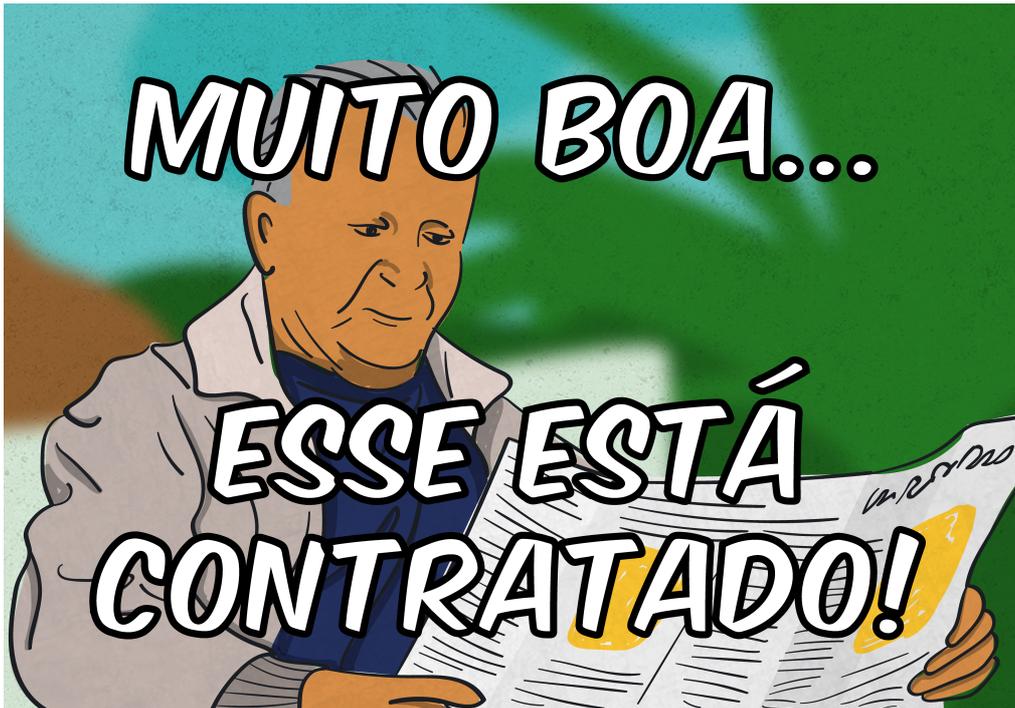
Os computadores são considerados muito rápidos em fazer as tarefas, sejam elas simples ou complexas, mas somente as fazem se tudo for dito corretamente, de forma clara e sempre na sequência correta. Assim, quando disser ao computador que ele deve realizar um procedimento, você precisa sempre seguir **uma sequência lógica de passos**, senão ao final o resultado poderá ser desastroso.

Veja o seguinte texto, no qual a mãe de um programador diz para ele ir ao supermercado realizar uma compra:

Figura 04 - A lógica no cotidiano do programador



Figura 05 - Contratação imediata



Fonte: Adaptado de <> Acesso em: 04 out. 2017.

Afinal, quem está correto? A mãe ou o programador? Ambos? O que causou o problema?

Isso ocorreu porque as instruções, que pareciam claras em um primeiro momento, não foram claras o suficiente para o programador. Ele comportou-se semelhantemente ao computador e seguiu “ao pé da letra” as instruções dadas pela mãe. E é sempre assim que os computadores trabalham: eles de fato obedecem às instruções que nós damos a eles.

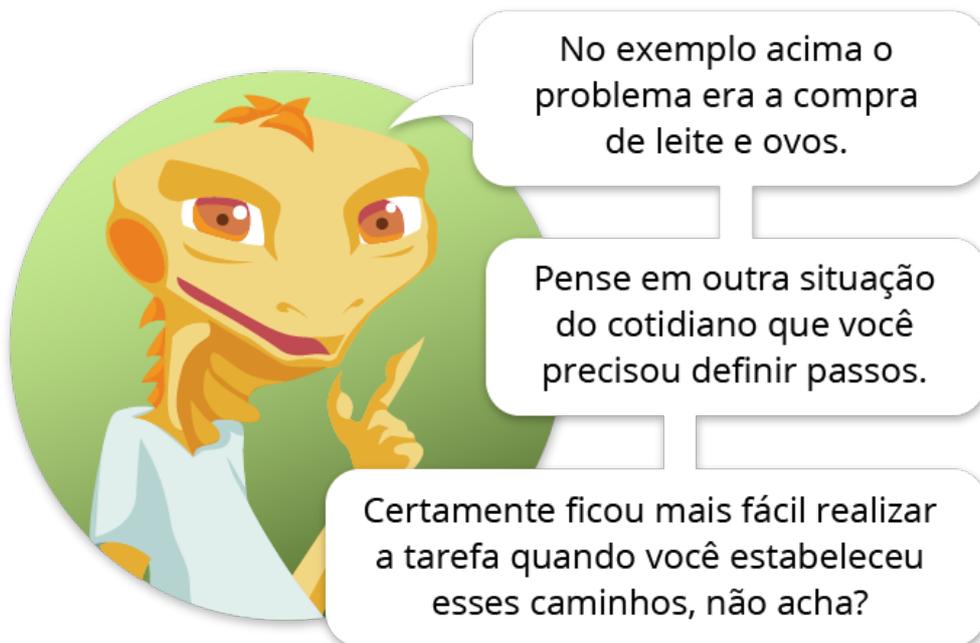
Lendo a tirinha acima, podemos dizer que somos capazes de entender que a mãe do programador solicitou a compra de um litro de leite e de seis ovos. No entanto, o programador não entendeu isso, pois faltou algo nas instruções e, assim, o resultado da ida ao supermercado foram seis litros de leite.

Como poderíamos reescrever as instruções da mãe para evitar o ocorrido?

Figura 06 - A lógica no cotidiano do programador



Repare que uma pequena informação adicionada às instruções modificou o resultado da ida ao supermercado. Por esse motivo, quando você estiver escrevendo as instruções para o computador, ou seja, os algoritmos, deve sempre prestar bastante atenção para fazê-los corretamente e com a máxima clareza. Caso contrário, o programa poderá apresentar um resultado diferente, como no exemplo citado, lhe dando “seis litros de leite”.



Quando tais instruções, também denominadas de comandos, não estão corretamente definidas, sem uma sequência lógica clara, o resultado apresentado não estará correto e, muitas vezes, pode nem se obter um resultado ao final dos passos realizados.

Como dar instruções ao computador?

Figura 07 - Linguagens de Programação



Quer saber o que foi dito ao computador? Para conseguir entender o código binário, você poderá [acessar aqui](#) e encontrar um tradutor capaz de convertê-lo para língua portuguesa.

A essa altura você já deve estar curioso para saber como conversar com o seu computador e dar-lhe as primeiras instruções, certo? Então, agora você conhecerá a língua utilizada pelos computadores. Lembre-se de que o computador é eficiente em fazer tarefas, mas, para isso, você deve lhe dizer que tarefas são essas, informando, na língua dele, cada passo a ser realizado.

A linguagem nativa dos computadores é a **binária**, a qual provavelmente você deve ter visto na disciplina Introdução às Tecnologias da Informação. Por meio dessa linguagem, eles entendem somente zeros e uns. Confesso que eu considero que

seria bem difícil aprender a se comunicar com os computadores utilizando apenas zeros e uns, mas não se preocupe, pois você poderá utilizar uma **Linguagem de Programação** para definir os comandos que deseja dar a essas máquinas.

Quando você utiliza uma **Linguagem de Programação**, os comandos que você escreve nela são convertidos para o código binário correspondente. Com esse procedimento, os computadores receberão as suas instruções na língua que eles entendem, ou seja, em binário. Esse processo de converter uma linguagem de programação em código binário é denominado **compilação**.

Você deve estar se perguntando agora: "Todas as Linguagens de Programação são compiladas?". A resposta para essa pergunta é: não! Algumas Linguagens de Programação têm os seus comandos convertidos para o código binário através de um **interpretador**. Nesses casos, os comandos escritos por você são convertidos em código binário apenas no momento de sua utilização. Na compilação é necessário converter os comandos para o código binário previamente. Resumindo, então, há as Linguagens de Programação que são interpretadas e há as que são compiladas.

Além dessa distinção, podemos identificar dois tipos de Linguagens de Programação: o de **Alto nível** e o de **Baixo nível**.

Figura 08 - Representação das linguagens de Alto e de Baixo Nível



O que isso significa?

- **Linguagens de Alto Nível:** são linguagens de programação que estão mais próximas da nossa língua, ou seja, estão mais distantes da linguagem da máquina. Java, Ruby e Python são exemplos de linguagens pertencentes à essa categoria.
- **Linguagens de Baixo Nível:** são linguagens de programação que possuem sua estrutura mais próxima da linguagem da máquina. Um exemplo de linguagem de baixo nível é o Assembly.

Veja os exemplos abaixo, os quais apresentam o programa “Olá Mundo” em duas linguagens: uma de **Alto Nível** e outra de **Baixo Nível**:

Na linguagem Potigol (Alto Nível):

```
1 escreva "Olá Mundo!"
```

Para utilizar essa linguagem não é necessário conhecer características específicas do processador, como instruções e registradores. Embora a Linguagem de Alto Nível seja mais compreensível, ela necessita ser precisa e sem ambiguidade. Há Linguagens de Alto Nível que são compiladas e há as que são interpretadas.

Na linguagem Assembly (baixo nível):

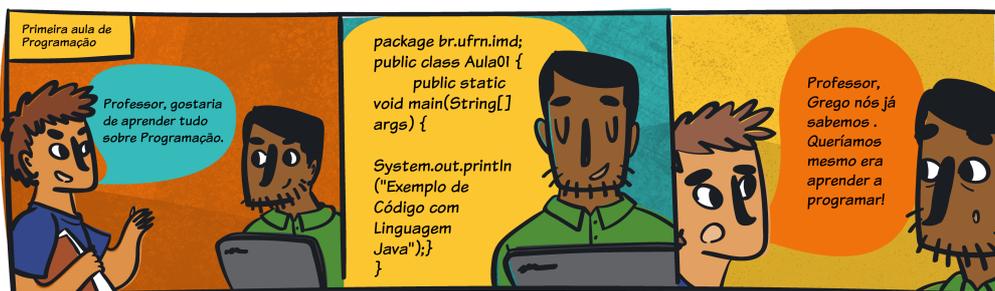
```
1 lea si, string
2 call printf
3 hlt
4 string db "Olá Mundo!" , 0
5 printf PROC
6     mov AL, [SI]
7     cmp AL, 0
8     je pfend
9     mov AH, 0EH
10    int 10h
11    inc SI
12    jmp printf
13    pfend:
14    ret
15 printf ENDP
16
```

As **linguagens Assembly** são diretamente relacionadas às características do processador ao qual se destinam. Quando um programa é criado com essas linguagens, o programador necessita conhecer diversos detalhes do processador, como os registradores e os comandos de operação. Um programa, chamado Montador (Assembler), transforma em código de máquina, isto é, em binário, as instruções escritas na Linguagem de Baixo Nível.

Entre os dois exemplos apresentados, a **Linguagem Potigol** pareceu muito mais amigável para você, não foi mesmo? Claro, um dos objetivos principais dessa linguagem é facilitar a vida do programador que, neste caso, é você. Já o objetivo do **Assembly**, que é uma Linguagem de Baixo Nível, é se comunicar melhor com a máquina.

As **Linguagens de Programação** são para os computadores o mesmo que os idiomas são para nós humanos. Nós somos capazes de falar os idiomas português, inglês ou qualquer outro, já os computadores são capazes de falar em Assembly, Java, Potigol, Python, etc.

Figura 09 - Na aula de programação



A essa altura você já deve conhecer um pouco do idioma inglês, inclusive você cursou a disciplina Inglês Técnico I e deve ter percebido que ele não é igual ao português, mas, ainda assim, usamos diversas palavras em inglês no dia a dia. Isto é, cada idioma possui estrutura e palavras próprias. Isso não é diferente com as linguagens de programação: cada uma delas também possui uma estrutura própria e suas palavras. A essa estrutura damos o nome de **sintaxe**.

Veja agora alguns pequenos trechos de código em algumas das linguagens mais populares:

Exemplo com linguagem Java

```
1 package br.ufrn.imd;
2 public class Aula01 {
3     public static void main(String[] args) {
4         System.out.println("Exemplo de Código com Linguagem Java");
5     }
6 }
```

Exemplo com linguagem C

```
1 #include <stdio.h>
2 int main(){
3     printf("Exemplo de Código com Linguagem C\n");
4     return 0;
5 }
```

Exemplo com linguagem Potigol

```
1 escreva "Exemplo de Código com Linguagem Potigol"
```

Você reparou como são diferentes essas linguagens de programação? Esses trechos de código realizam a mesma ação: imprimem uma mensagem na tela. Os exemplos apresentados são apenas para lhe mostrar que existem diferenças entre as diversas linguagens existentes.

A solução de um problema pode ser descrita através de um **algoritmo**. Vou lhe dar um exemplo de algo que você faz em seu cotidiano: após tomar banho, você precisa executar uma sequência lógica para que, ao final, esteja vestido para ir à escola ou ao encontro presencial do IMD. Observe o algoritmo abaixo:

1. Enxugar-se;
2. Vestir as roupas de baixo;
3. Vestir a calça jeans;
4. Calçar as meias;
5. Calçar o tênis;
6. Vestir a camiseta da farda.

Se você trocar a ordem de alguns dos passos, poderá até estar vestido ao final, mas será que é interessante colocar as meias após calçar o tênis? Ou, então, vestir a roupa de baixo após vestir a calça jeans? Acredito que vai ficar estranho em qualquer uma dessas duas situações.

E aí, você acha que esse exemplo se aplicaria ao Superman?

Figura 10 - Superman



Fonte: PIN ART. Disponível em: <>. Acesso em: 09 out. 2017.

Não mesmo, rrsrsrs... Para ele, você tem de ter um algoritmo bem específico.

Escolhi lhe dar esse exemplo para demonstrar que o algoritmo define os passos a serem seguidos visando alcançar, como resultado, a solução de um problema e garantindo que, sempre que executado sob as mesmas condições, produza o

mesmo resultado. Assim, podemos dizer que **programar é construir algoritmos utilizando uma linguagem de programação como ferramenta.**



Curiosidade

Você sabia que o primeiro algoritmo conhecido foi construído por uma programadora? Em meados do século XIX, a Condessa de Lovelace, conhecida como Ada Lovelace, tornou-se a primeira programadora da história ao criar o primeiro algoritmo para ser processado por uma máquina.

Quer saber um pouco mais sobre a história de Lovelace? [Clique aqui.](#)

Os algoritmos possuem ainda: entrada, processamento e saída. Esses elementos são assim definidos:

- **Entrada:** são os dados informados ao programa para ele ser executado.
- **Processamento:** é a sequência de passos executados para se alcançar a solução de um problema.
- **Saída:** diz respeito aos dados gerados pelo processamento realizado pelo programa.

Agora faça um algoritmo em que são necessárias informações de entrada. Que tal fazer um pudim? Digo, o algoritmo “Receita de Pudim”? Você vai precisar das informações de entrada, ou seja, dos ingredientes; realizar o processamento, que é o preparo do pudim; e terá a saída do algoritmo, o próprio pudim. Vamos lá, então!?

Digite aqui sua resposta!

Você conseguiu entender como preparar um algoritmo? Eu falei algoritmo, viu? Espero que não tenha se distraído com o pudim... Para que seu algoritmo fique mais claro, recomendo seguir as seguintes técnicas:

- Cada passo deve possuir apenas um verbo;
- Usar frases curtas, simples e objetivas;
- Escrever frases que não tenham sentido dúbio.

Acredito que agora você já sabe como escrever seu primeiro algoritmo. O que acha de tentar? Caso tenha dúvidas, consulte o seu professor mediador para ele poder auxiliá-lo com os exercícios.



Atividade 01

1. Escreva o algoritmo necessário para organizar o seu quarto e o de seu(s) irmão(s)/irmã(s).
2. Escreva os passos realizados para um algoritmo que realizará a instalação de um programa ou de um jogo em seu computador.
3. Escreva detalhadamente os passos realizados para um algoritmo conseguir calcular a média final dos alunos do Curso Técnico em TI do IMD/UFRN. Você poderá consultar o Manual do Aluno para ajudá-lo nessa tarefa.

Agora que você já sabe estruturar seus algoritmos, vá para o próximo passo: conheça a linguagem de programação a ser utilizada para dizer ao computador a tarefa que ele deve realizar! Para isso, você utilizará a **linguagem Potigol**, que é uma linguagem de programação em português, voltada para iniciantes. Devido a seus

comandos serem em português, essa é uma linguagem de fácil entendimento, por meio da qual você pode focar no aprendizado de suas habilidades de lógica de programação.

Veja o exemplo de código com o Potigol:

```
1 escreva "Exemplo de Código com Linguagem Potigol"
```

Esse exemplo exibe na tela o texto “Exemplo de Código com Linguagem Potigol”. Quando você o compara com os códigos das linguagens C e Java, percebe sua simplicidade. Isso lhe permite focar na construção dos algoritmos.

A partir desse momento você poderá exercitar os seus algoritmos no material didático. Na próxima página você conhecerá o recurso de correção de algoritmos e, caso você fique com dúvidas, peça ao seu professor mediador para que ele lhe auxilie.



Atividade 02 - Olá Mundo!

Conteúdo interativo, acesse o Material Didático.



Resumo

Nesta aula você viu os conceitos relativos à lógica de programação, entendeu a importância da definição de comandos claros e a maneira como se pode criar os primeiros algoritmos. Além disso, conheceu o recurso que será utilizado para testar os algoritmos escritos com o Potigol.

Criar algoritmos pode parecer bem mais difícil do que executar receitas, porém, no decorrer das aulas e no desenvolvimento da disciplina, você irá não só se familiarizar mais com eles, como também compreender bem melhor o assunto. Conte comigo e com seus professores mediadores para quaisquer esclarecimentos de dúvidas. Por hoje, fico por aqui... Até a próxima aula!



Referências

ASCII Text to Binary Converter. Disponível em: <>. Acesso em: 25 de set. de 2017.

CHAUÍ, M. **Convite à filosofia**. 13. ed. São Paulo: Ática, 2003.

FORBELLONE, André Luiz Villar. **Lógica de programação: a construção de algoritmos e estruturas de dados**. São Paulo: Prentice Hall, 2005.

GNIPPER, Patrícia. Mulheres históricas: Ada Lovelace, a primeira programadora de todos os tempos. Disponível em: <>. Acesso em: 25 set. 2017.

MORAES, Paulo Sérgio de. **Lógica de programação**. Unicamp - Centro de Computação - DSC, 2000. Disponível em: <>. Acesso em: 11 set. 2017.

POTIGOL é... Disponível em: <>. Acesso em: 12 set. 2017.

SILVA, Erbi. **Que linguagens usam os sites mais conhecidos?**. Disponível em: <>. Acesso em: 25 set. 2017.

SCUDERO, Erick. **Linguagens de alto nível vs. baixo nível: qual é melhor?** Disponível em: <>. Acesso em: 25 set. 2017.

