

Coleção Livro Aberto de Matemática
Ensino Médio

Pensamento Computacional

Livro do Aluno



Leonardo Barichello



Pensamento Computacional

2ª edição, 9 de agosto de 2023

Leonardo Barichello

ISBN: 9780-85-224-0524-2

Publicado no Brasil / Published in Brazil

Licença



Coleção Livro Aberto de Matemática

Revisão: Dênis Vargas, Letícia Rangel, Diego Lieban, Larissa Monzon, Kátia Rocha

Capa: Enzo Esberard

Ilustrações: Agnes Antonello

Diagramação e Gráficos: Tarso Boudet Caldas

Realização: Instituto de Matemática Pura e Aplicada (IMPA)

Estrada Dona Castorina, 110

Jardim Botânico

22460-320, Rio de Janeiro, RJ

www.impa.br

editora@impa.br

B2521 Barichello, L.,
Pensamento Computacional/Leonardo Barichello — 2ª ed. — Rio
de Janeiro, IMPA, 2023.
32 p.: il. color (Coleção Livro Aberto de Matemática)

E-book

ISBN 9780-85-224-0524-2 (Versão aluno)

1. Matemática. I. Título II. Série

CDD: 510
Carolina Celano Lima/CRB-7: 2438

SUMÁRIO

| | |
|--|-----------|
| 1 Um novo jeito de resolver problemas | 1 |
| Explorando: Um novo jeito de resolver problemas | 1 |
| Organizando: Um novo jeito de resolver problemas | 3 |
| Explorando: A representação via fluxogramas | 4 |
| Organizando: A representação via fluxogramas | 5 |
| Para Saber +: Instruções para um computador | 6 |
| 2 Variáveis e operações | 9 |
| Explorando: Variáveis e operações | 9 |
| Organizando: Variáveis e operações | 11 |
| 3 Condicionais | 13 |
| Explorando: Condicionais | 14 |
| Organizando: Condicionais | 14 |
| 4 Repetições | 19 |
| Explorando: Repetir e repetir | 20 |
| Organizando: Repetir e repetir | 22 |
| 5 Praticando tudo que foi estudado | 25 |
| Praticando: Tudo o que você aprendeu | 25 |
| Referências | 28 |

O quê?

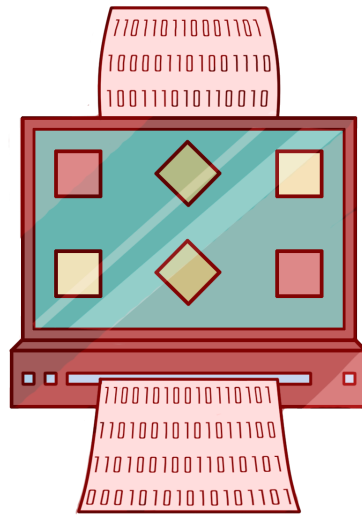
Neste material trataremos de Pensamento Computacional através da resolução de problemas matemáticos por meio do uso de linguagens de programação.

Por quê?

Os computadores estão presentes em quase todas as atividades humanas na atualidade, seja nos mais diversos ramos profissionais, nas nossas atividades de lazer, quando interagimos com outras pessoas e na produção de conhecimento científico. Por isso, desenvolver habilidades relacionadas ao uso de softwares e computadores, aos fundamentos por trás do seu funcionamento e ao impacto do uso dessas tecnologias na nossa vida é fundamental. Pensamento computacional é o nome dado a uma parte dessas habilidades, mais especificamente, às habilidades relacionadas à resolução de problemas a partir de técnicas, conceitos e ferramentas típicos da Ciência da Computação. Essas habilidades vão te ajudar a conhecer noções básicas de linguagens de programação, o que é cada vez mais importante para sua vida profissional, trajetória acadêmica e participação efetiva na sociedade.

1 UM NOVO JEITO DE RESOLVER PROBLEMAS

Os computadores estão em todos os lugares, seja na forma de celulares, notebooks, computadores de mesa ou dispositivos com funções mais específicas, como máquinas de cobrança. Das atividades mais corriqueiras às atividades mais inovadoras, é possível identificar a influência de um computador, e na matemática não é diferente.



Nas últimas décadas, a comunidade de matemáticos teve que aprender a lidar com o uso intensivo de computadores em uma área do conhecimento que, por muito tempo, aceitava de forma quase única argumentos lógicos na forma textual. Um exemplo ocorreu com o conhecido **problema das quatro cores** que, em resumo, lançava a questão sobre quantas cores são necessárias para pintar um mapa sem que países vizinhos tenham a mesma cor.

Em 1852, uma resposta para esse problema foi sugerida por um matemático: 4 cores seriam suficientes, mas ele não demonstrou que essa resposta de fato estava correta. Mais de 100 anos se passaram até que um grupo de matemáticos conseguiu demonstrar que a tal resposta estava de fato correta, mas a demonstração era feita com o auxílio de computadores e tomava mais de mil horas de processamento nos computadores mais rápidos existentes naquele momento.

O que incomodou os matemáticos não foi apenas o tempo quase sobrehumano para checar essa solução, mas principalmente o questionamento sobre a validade ou não de uma demonstração que dependia de um computador para ser realizada.



Você Sabia?

Você pode saber mais sobre o problema das quatro cores assistindo ao vídeo disponível em youtu.be/TsAuQkbkW50.

O que essa história ilustra é que a disponibilidade de computadores faz mais do que agilizar certas tarefas repetitivas, mas cria a possibilidade de resolvermos problemas de maneira diferente do que poderíamos sem eles. A capacidade de utilizar um computador amplifica as nossas possibilidades como resolvedores de problemas.

Neste material, vamos desenvolver o seu pensamento computacional partindo de problemas matemáticos como os que você já está acostumado a resolver e buscar chegar até a criação de algoritmos que resolvam esses problemas e que possam ser executados por um computador!



Atividade 1

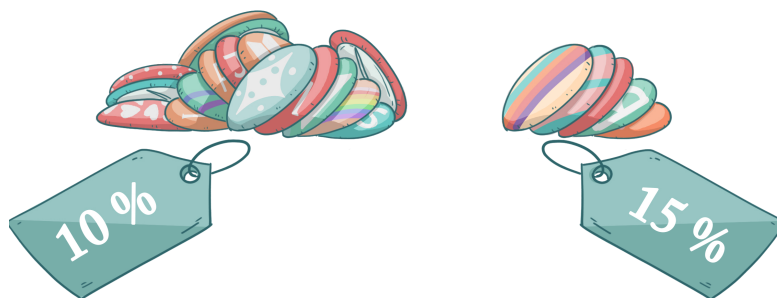
Boas instruções

O proprietário de uma empresa que fabrica e vende *bottons* gostaria de criar uma promoção para incentivar a compra de grandes quantidades, uma vez que isso agiliza a produção. Na sua loja, todos os *bottons* são vendidos pelo mesmo preço: R\$ 2,00.

A primeira ideia foi oferecer um desconto de 10% no valor total da compra caso o comprador adquirisse mais do que 100 unidades.

- a) Com essa promoção, qual seria o valor de uma compra de 95 *bottons*? E de uma compra de 105 *bottons*?

Como você deve ter notado, se essa proposta de desconto for utilizada compras com poucas unidades acima de 100 podem ficar mais baratas do que compras que não qualificaram para o desconto.



Uma outra ideia do proprietário foi a seguinte: cada unidade além da centésima recebe 15% de desconto no seu valor. Por exemplo, se alguém comprar 105 *bottons*, o comprador paga o preço normal para 100 deles e depois recebe 15% de desconto no valor dos outros 5.

- b) Com essa promoção, qual seria o valor de uma compra de 95 *bottons*? E de 105? E de 200 *bottons*?
- c) Você acha que essa ideia evita o problema de compras com pouco mais de 100 *bottons* ficarem mais baratas do que compras com quantidades menores? Justifique a sua resposta.

O proprietário da empresa decide adotar essa segunda ideia, mas notou que ela cria um problema: o cálculo do valor final da compra não pode mais ser feito de maneira imediata na calculadora.

- d) Descreva com as suas palavras como um funcionário deve proceder para calcular o valor de uma compra a partir da informação de quantos *bottons* foram comprados usando uma calculadora simples. Dê a sua resposta na forma de um bilhete que será lido por uma pessoa que você não irá encontrar pessoalmente.
- e) Troque o seu bilhete com um colega e discuta com ele se vocês seriam capazes de compreender como o preço de uma compra é calculado se vocês tivessem apenas lido o bilhete escrito por cada um.



Uma resolução boa de repetir

O cometa Halley é um dos cometas de menor período do Sistema Solar, completando uma volta em torno do Sol a cada 76 anos. Na última ocasião em que ele ficou visível do planeta Terra, em 1986, várias agências espaciais enviaram sondas para coletar amostras de sua cauda e assim confirmar teorias sobre suas composições químicas.

Figura 1: Registro da passagem do cometa Halley em 1682



Fonte: Doolittle (1910)

- Uma pessoa que nasceu em 2020 vai ter a oportunidade de avistar o cometa Halley pela primeira vez em que ano? E uma que venha a nascer no ano 2200? E no ano 3000?
- Qual foi o ano em que o cometa Halley pode ter sido avistado pela primeira vez para uma pessoa que nasceu em 1900? E uma que nasceu em 1500?
- Em uma folha de papel à parte, descreva como obter a resposta para questões como as anteriores para um amigo fictício que tenha terminado o Ensino Médio, mas não tenha resolvido essas questões. Tenha em mente que o objetivo não é explicar como resolver o problema, mas descrever um procedimento que permita ao seu colega obter respostas. Use texto, expressões matemáticas, esquemas visuais ou outros recursos que possam ajudar a deixar as suas instruções o mais claras possível.
- Troque as instruções com um colega, e seguindo os passos que ele descreveu, obtenha o ano do primeiro possível avistamento do cometa Halley para alguém que tenha nascido nos anos: 2500, 1000 e 2290.
- Antes de devolver a resolução para o seu colega, avalie-a considerando os seguintes aspectos: Ele seguiu a mesma estratégia que você? Na sua opinião, qual dos dois conjuntos de instruções é mais fácil de seguir e porquê?

As atividades anteriores mostraram que não é simples criar um conjunto de instruções que resolve um determinado problema de modo que uma pessoa seja capaz de compreendê-las e segui-las sem precisar fazer mais perguntas ou pedir esclarecimentos. Por isso é importante que fique claro tanto o significado de cada instrução, quanto a ordem em que elas devem ser executadas.

No caso de criarmos instruções para um computador seguir, essa exigência fica ainda maior, pois o computador não consegue usar bom senso ou conhecimentos adicionais sobre a situação para interpretar as instruções ou tomar decisões que não estejam claras, caso isso seja necessário. Para um computador, todas as instruções dadas devem ser conhecidas por ele e o fluxo, ou seja, a ordem em que elas devem ser executadas, deve ser absolutamente claro na descrição.

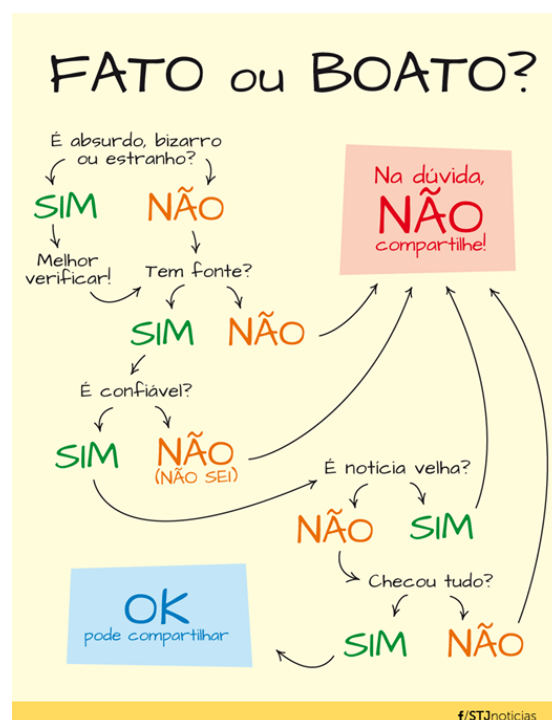
Um conjunto finito de instruções bem definidas e com fluxo claro é chamado de algoritmo.

Ao longo deste material, vamos conhecer várias ferramentas que nos permitam representar um conjunto de instruções. Em última instância, estamos procurando desenvolver habilidades e conhecer ferramentas que nos permitam criar algoritmos, e algoritmos podem ser usados para criar programas e aplicativos que realizem certas tarefas para nós!

Uma dessas formas de representação é o **fluxograma**. Trata-se de uma forma de representar visualmente a maneira como as instruções (ou comandos) se relacionam e qual a ordem em que devem ser realizadas (ou executados).

Você já deve ter visto esquemas visuais como o mostrado a seguir. Este esquematiza o processo de decisão sobre a pertinência de compartilhar uma notícia.

Figura 2: Campanha do Superior Tribunal de Justiça (STJ)



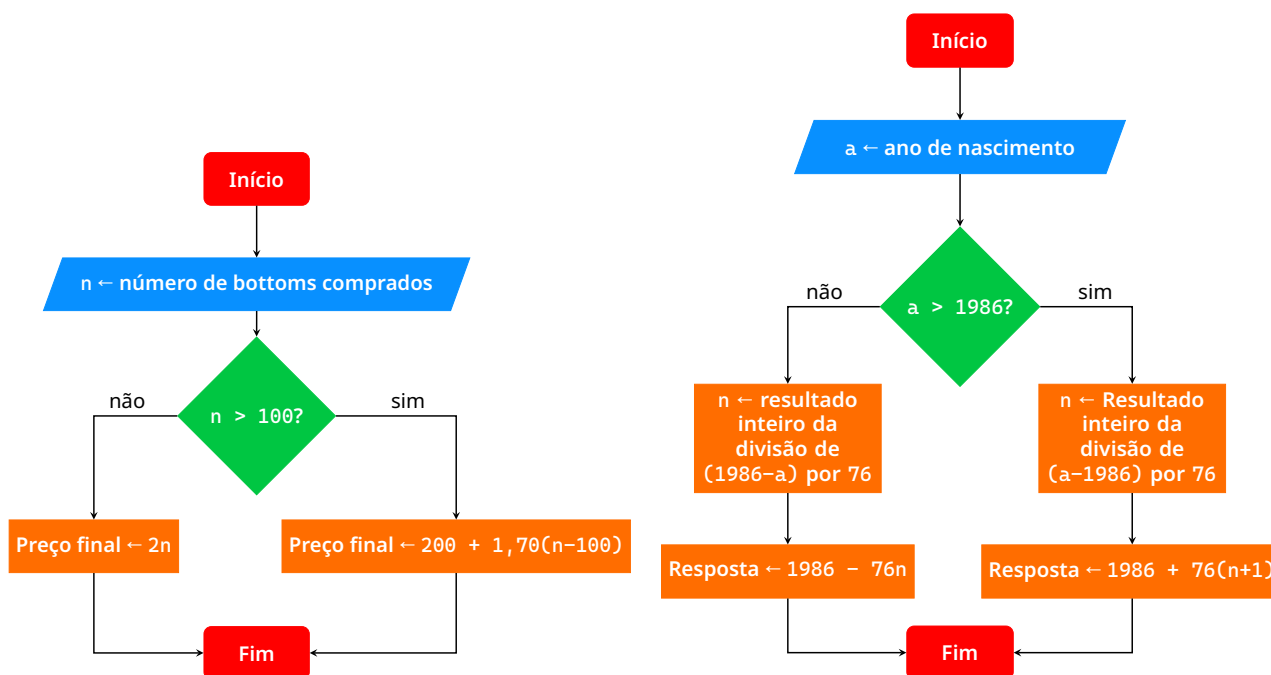
Fonte: Página do Facebook do STJ

Em algumas atividades deste material, faremos um uso mais específico e rigoroso desse recurso, mas a intenção é a mesma da imagem anterior: representar visualmente um processo com múltiplas etapas e ações.

Atividade 3

Usando fluxogramas

A imagem a seguir mostra dois fluxogramas que resolvem as atividades anteriores. Leia-os com atenção e, se necessário, volte as [Atividades 1](#) e [2](#) para relembrar.



- Utilize o fluxograma da esquerda para determinar qual seria o valor de uma compra de 60 e outra de 150 bottoms
- Utilize o fluxograma da direita para determinar em que ano o cometa Halley poderá ser avistado pela primeira vez por alguém que tenha nascido no ano de 2300.

Organizando

A representação via fluxogramas

Os fluxogramas são uma maneira de representar algoritmos que fazem uso de um arranjo visual para deixar claro o fluxo, ou seja, a sequência em que comandos devem ser realizados. Embora seja conveniente em muitas situações, eles não são recomendados para algoritmos muito complexos ou longos.

Mesmo assim, eles ainda podem ser uma forma rápida para organizar o seu raciocínio quando estiver pensando em um algoritmo. Por isso, consideramos que vale a pena mostrar mais um exemplo que ilustra uma situação não mostrada anteriormente: um processo repetitivo.

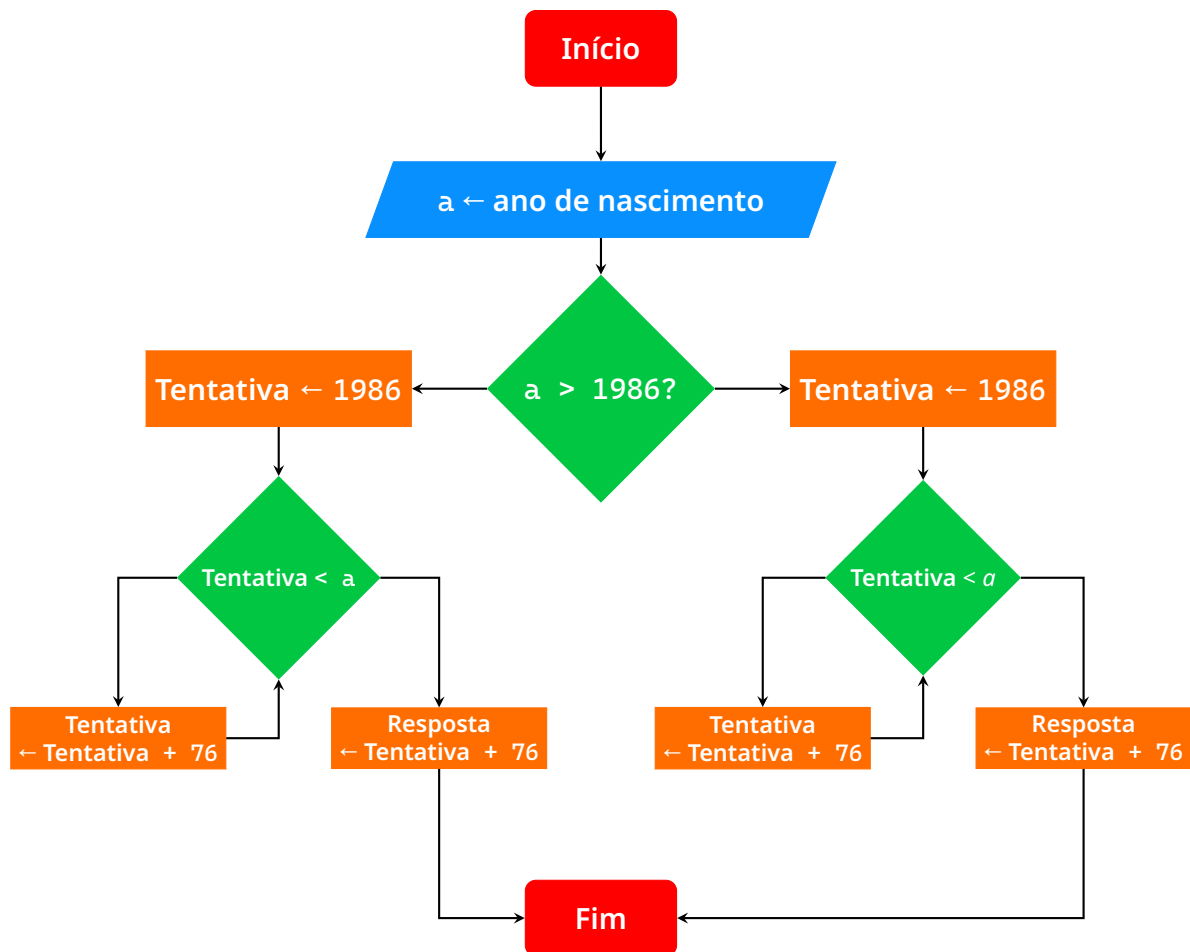
Considere a seguinte solução (descrita na forma textual) para a [Atividade 2](#):

Se o ano de nascimento for maior que 1986, então some 76 a 1986 até que o resultado seja maior ou igual ao ano de nascimento. Quando isso ocorrer, o resultado da última soma é a resposta. Se o



ano de nascimento for menor do que 1986, então subtraia 76 a 1986 até que o resultado seja menor que ano de nascimento. Quando isso ocorrer, some 76 ao resultado da última soma e essa será a resposta.

Como fluxograma, ela pode ser representada como mostrado a seguir.



Dois detalhes merecem atenção neste fluxograma.

O primeiro deles se refere aos termos **tentativa**, **a** e **resposta**. Concretamente, podemos pensar neles como nomes que damos a certos valores que serão usados ao longo na nossa resolução. Em computação, esses termos são chamados de **variáveis** e representam espaços na memória do computador que foram reservados para o seu algoritmo armazenar alguma informação. Vamos considerar o caso do variável **tentativa**. Ela é usada para armazenar em qual ano estamos à medida que somamos ou subtraímos 76 até encontrar a resposta. O ato de armazenar uma informação em uma variável nesse fluxograma é representada por \leftarrow , como em $tentativa \leftarrow tentativa + 76$. Esse comando diz ao computador para calcular o valor do lado direito de \leftarrow (somando 76 ao valor atual da variável **tentativa**) e então guardar na variável indicada à esquerda do \leftarrow (**tentativa**) o resultado. Com isso, o resultado do cálculo é escrito “em cima” do valor que estava armazenado até antes da execução desse comando na variável em questão.

O segundo detalhe é a ocorrência de ciclos: esses ciclos representam processos iterativos, ou seja, que envolvem repetição sucessiva, são repetitivos. Focando no ciclo que está mais à esquerda, ele diz o seguinte: cheque se **tentativa** é menor do que **a**, se não for, subtraia 76 e volte para a checagem. Isso faz com que o algoritmo repita essa subtração até que o resultado encontrado seja menor do que **a**. No ciclo do lado direito, ocorre o mesmo mas com adições (veja que o lado direito do fluxograma é acionado quando **a** é maior do que 1986).

Processos repetitivos são muito comuns em algoritmos e voltaremos a eles em breve.



Para que um computador seja capaz de seguir instruções que realizem alguma tarefa, é necessário que o algoritmo seja escrito em uma linguagem de programação. Uma linguagem de programação nada mais é do que uma maneira muito estruturada de descrever um algoritmo. A necessidade de clareza e estrutura para que um computador consiga interpretar um algoritmo faz com que certos elementos aparentemente estranhos sejam usados para organizar o fluxo das instruções.

Existem centenas de linguagens de programação em uso na atualidade, cada uma com certas vantagens e desvantagens e com diferentes níveis de popularidade. Atualmente, Python é uma linguagem muito utilizada em vários contextos e conta com muitos materiais para autoestudo na internet. Javascript é um outro exemplo, mas seu uso é mais comum em páginas de internet. A linguagem C é muito usada, mas normalmente para programas mais técnicos que precisam de alto rendimento.

A imagem a seguir mostra dois algoritmos escritos em uma linguagem de programação chamada **Portugol**, que tem seus comandos em português. Apesar de não ser usada comercialmente, essa é a linguagem que sugerimos para você neste capítulo.

As instruções descritas abaixo em Portugol são as mesmas descritas anteriormente com fluxogramas na [Atividade 3](#). Leia com calma e tente compreender como as representações se relacionam.

```
1 programa {
2   funcao inicio() {
3     inteiro n
4     real preco
5     leia(n)
6     se (n>100) {
7       preco = 200 + (n-100)*1.70
8     }
9     senao {
10      preco = 2*n
11    }
12    escreva(preco)
13  }
14 }
```

```
1 programa {
2   funcao inicio() {
3     inteiro a, n, resposta
4     leia(a)
5     n = (a-1986) / 76
6     se (a>1986) {
7       n=n+1
8     }
9     resposta = 1986 + n*76
10    escreva(resposta)
11  }
12 }
```

Embora não seja necessário entender todos os detalhes que fazem partes desse algoritmo escrito em Portugol, como o uso dos símbolos { e }, um detalhe é importante: o uso do símbolo =.



Em Portugol, assim como na maioria das linguagens de programação, o símbolo = não é usado para expressar uma igualdade entre dois termos (como em uma equação), mas sim para representar a ação de atribuição: a variável à esquerda do sinal deve receber o valor à direita. Esse é o significado do símbolo ← nos fluxogramas que discutimos na atividade anterior.

Por causa disso, é comum vermos expressões como “ $t=t+76$ ” em algoritmos escritos em linguagens de programação.

O sinal de igual

Observação

Em um contexto matemático, o sinal de igual nessa expressão significaria uma equivalência entre os valores do seu lado direito e esquerdo. Nesse caso específico, a expressão é uma equação sem solução (pois não existe valor de t que satisfaça a igualdade proposta).

Em linguagens de programação, essa expressão significa “calcule o valor da expressão ao lado direito do símbolo = e armazene na variável indicada no lado esquerdo”.

Por isso, tenha muito cuidado com o contexto ao usar o símbolo =, pois ele pode ser interpretado de maneira incorreta se o contexto não estiver claro.

Ao longo das próximas atividades vamos incentivar o uso de Portugol para escrever algoritmos e, pouco a pouco, vamos esclarecer o significado de todos os elementos que apareceram nas imagens anteriores.

O Portugol Studio é um software que interpreta algoritmos escritos em Portugol e pode ser instalado no seu computador (<http://lite.acad.univali.br/portugol>) ou no seu celular. Você também pode acessá-lo diretamente do navegador em <http://portugol-webstudio.cubos.io>. Caso você não esteja familiarizado com esse tipo de recurso, sugerimos o vídeo <http://youtu.be/6OIADpFImtc> como ponto de partida.



2 VARIÁVEIS E OPERAÇÕES

Dois recursos são muito comuns na criação de algoritmos, não importando se descrevemos na forma textual, com fluxograma ou em uma linguagem de programação: condicionais e repetições. Além desses dois recursos, o conceito de variável é especialmente importante quando pensamos em linguagens de programação.

Já conhecemos esses três elementos nas atividades anteriores, mas nas atividades a seguir vamos ter a chance de explorá-los com mais profundidade.

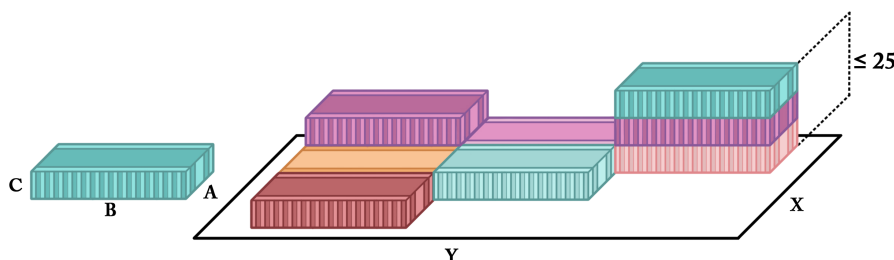
Explorando

Variáveis e operações

Atividade 4

Contêineres

O transporte em grandes navios de carga se dá através de contêineres, que são caixas metálicas grandes dentro das quais as empresas acomodam seus produtos. A maneira como os contêineres são colocados nos navios costuma ser determinada pelo sistema de carregamento disponível nos portos. Imagine um porto que, por restrição nos equipamentos disponíveis, carregue seus contêineres sempre alinhados da mesma forma, como mostrado na figura abaixo.



A administradora desse porto está com problemas para determinar quantos contêineres de dimensões A , B e C (em metros) podem ser colocados em um navio que tenha área de carregamento nas dimensões X e Y (em metros). Observe que a dimensão A dos contêineres deve ser carregada paralelamente à dimensão X no navio, o mesmo ocorre para as dimensões B e Y .

Além dessas restrições, há uma limitação de altura que diz que a pilha de contêineres não pode ultrapassar 25 metros de altura.

- Quantos contêineres de dimensão (em metros) $A = 3$, $B = 4$, $C = 1$ cabem em um navio com área de carregamento com dimensões $X = 30$ e $Y = 60$?
- Quantos contêineres de dimensão (em metros) $A = 4$, $B = 8$, $C = 2$ cabem em um navio com área de carregamento com dimensões $X = 30$ e $Y = 60$?
- Descreva, com suas palavras, como determinar quantos contêineres de dimensões A , B e C cabem em um navio com espaço de carregamento igual a X e Y .



Um professor em cada van

Uma escola costuma organizar passeios com frequência e para levar os alunos conta com vários motoristas de vans. Essas vans possuem 14 lugares, além do assento do motorista. Por questão de segurança, a direção da escola exige que sempre haja um professor em cada van, não importando o número de alunos. Felizmente, a escola tem contato com muitos motoristas de vans.



- Quantos professores serão necessários para acompanhar os estudantes em um passeio em que 200 estudantes desejem participar? E em um passeio em que 65 estudantes desejem participar?
- Descreva com suas palavras, e de forma clara, como obter a quantidade mínima de professores necessários para acompanhar um passeio em que n estudantes estejam interessados em participar.

Em algumas ocasiões, porém, há mais professores dispostos a acompanhar os estudantes do que o necessário. Em geral, a direção gosta dessa situação pois cada professor fica responsável por menos estudantes. Porém, já aconteceu de não haverem lugares disponíveis nas vans para os professores adicionais, e a escola não pretende contratar vans adicionais.

- Se houverem 200 estudantes interessados e 20 professores disponíveis, haverá lugar para todos os professores nas vans sem que seja necessário contratar vans adicionais? E se forem 75 estudantes e 10 professores?
- Descreva com suas palavras e de forma clara como descobrir o número máximo de professores que podem acompanhar n estudantes em um passeio sem que seja necessário contratar mais vans.



Para resolver as duas atividades anteriores você deve ter utilizado apenas algumas operações: soma, subtração, multiplicação, divisão e arredondamentos. As quatro primeiras estão diretamente disponíveis em qualquer linguagem de programação e são suficientes para resolver a [Atividade 4](#). Vejamos como seria o algoritmo para resolvê-la em Portugol:

```
1 programa {
2   funcao inicio() {
3     inteiro a, b, c, x, y, capacidade
4     leia(a)
5     leia(b)
6     leia(c)
7     leia(x)
8     leia(y)
9     capacidade = (x/a)*(y/b)*(25/c)
10    escreva(capacidade)
11  }
12 }
```

Veja que na linha 9 é realizada a divisão de cada uma das dimensões do navio pelas dimensões de cada contêiner para determinar quantos contêineres cabem em cada dimensão e, então, as quantidades são multiplicadas para determinar a capacidade do navio.

Vale a pena salientar um aspecto sobre o cálculo do linha 9 acima: como estamos lidando com variáveis inteiras, o Portugol entende que a divisão a ser realizada também é uma divisão inteira, ou seja, sem parte decimal no quociente. Assim, o resultado de x/a é a quantidade (inteira) de contêineres que cabem na dimensão x do navio, como queríamos.

Já na [Atividade 5](#), você deve ter precisado de uma ação além das quatro operações: arredondamento. O recurso de arredondar para cima é necessário para descobrir o número mínimo de professores necessários para acompanhar a turma.

Por exemplo, se 30 alunos quiserem participar de uma viagem, calculamos $30/13 \approx 2,3$. Logo, precisamos de mais de 2 professores, ou seja, 3. Matematicamente, dizemos que obtivemos o menor número inteiro maior ou igual ao resultado da divisão. Informalmente, dizemos que estamos “arredondando para cima”. Porém, o Portugol (e outras linguagens de programação) não entende a instrução “arredonde para cima” ou “obtenha o menor número inteiro maior ou igual”. Então, como poderíamos descrever para o computador esse processo?

Para fazer isso, podemos utilizar duas operações que a maioria das linguagens de programação oferece: quociente da divisão (representada por $/$ e resto da divisão entre dois números inteiros (representada por $\%$).

- A operação $/$ retorna um número real igual ao quociente da divisão se os valores envolvidos forem números reais, por exemplo, $7.2/4.8$ terá como resultado 1.5. Caso os valores envolvidos sejam números inteiros, ela retorna a parte inteira do quociente, por exemplo, $9/4$ terá como resultado 2.
- Já a operação $\%$ só pode ser realizada entre dois números inteiros e retorna o resto da divisão, por exemplo, $9\%4$ terá como resultado 1.

No caso do problema posto na atividade, devemos adicionar uma van se a divisão do número de interessados por 13 tiver resto maior do que zero. Em Portugol, uma forma de resolver o problema é essa:



```

1 programa {
2   funcao inicio() {
3     inteiro n, interessados, maximoprofs
4     leia(interessados)
5     n = interessados/13
6     se (interessados%13 > 0) {
7       n=n+1
8     }
9     maximoprofs = 14*n-interessados
10    escreva("São necessários no mínimo ", n)
11    escreva(" e podem ir no máximo ", maximoprofs)
12  }
13 }

```

Veja que esse algoritmo primeiro considera que o número mínimo de professores (n) é igual ao resultado inteiro da divisão do número de interessados por 13. Depois, se o resto da divisão por 13 for maior do que zero, o algoritmo soma 1 ao número mínimo de professores.



Para refletir

Agora que você sabe como realizar algumas operações em Portugol, você já consegue criar algoritmos que realizem cálculos matemáticos simples a partir de um conjunto de valores dados. Você pode, por exemplo, criar um algoritmo que leia três notas e calcule a média entre elas. Que tal tentar?



3 CONDICIONAIS

Condicionais são um componente central em qualquer linguagem de programação e, na verdade, em todos os recursos computacionais que utilizamos. Elas servem para que o computador realize ações diferentes de acordo com o caso com que estiver lidando. Por exemplo, na [Atividade 1](#) haviam dois casos que exigiam um conjunto de comandos diferentes para que se calculasse o preço final da compra: o caso com até 100 bottons e o caso com mais de 100 bottons.

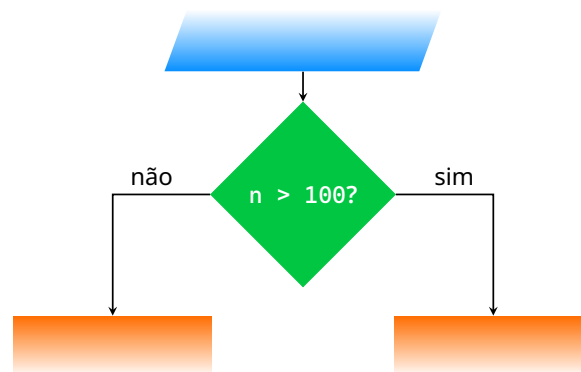
No caso das planilhas eletrônicas, por exemplo, elas podem ser usadas para que uma célula mostre um conteúdo que depende do conteúdo de uma outra célula. Um exemplo simples é mostrado a seguir. Imagine que um professor tenha uma planilha em que registra as notas dos seus estudantes em duas provas. Além de calcular a média das duas notas, ele deseja que a planilha mostre se cada estudante está, ou não, de recuperação de acordo com o seguinte critério: se a média for menor do que 6 então o estudante está de recuperação, senão está aprovado.

| | A | B | C | D |
|---|----------------|----------------|--------------|--------------------------------------|
| 1 | Prova 1 | Prova 2 | Média | Situação |
| 2 | 6,5 | 4,5 | $=(A2+B2)/2$ | $=SE(C2<6;"Recuperação";"Aprovado")$ |

| | A | B | C | D |
|---|----------------|----------------|--------------|-----------------|
| 1 | Prova 1 | Prova 2 | Média | Situação |
| 2 | 6,5 | 4,5 | 5,5 | Recuperação |

Note que o comando descrito na frase “se a média for menor do que 6, então o estudante está de recuperação, senão está aprovado” é composto por três partes: uma condição ($C2<6$), o que deve ser feito se a condição for verdadeira (escrever a palavra “Recuperação” na célula) e o que deve ser feito se a condição for falsa (escrever “Aprovado” na célula). Isso está condensado no conteúdo da célula D2 mostrada anteriormente.

Nas atividades anteriores, usamos o mesmo raciocínio para resolver o problema do cometa Halley (se o ano de nascimento fosse maior do que 1986 a resolução seguia por um caminho, se fosse menor seguia por outro) e o problema sobre o número de professores (se houvesse resto na divisão, era necessário contratar mais uma van). Quando representamos um algoritmo na forma de um fluxograma, condicionais são representadas como bifurcações no fluxo das instruções, de modo que um ou outro lado deve ser seguido conforme a condição estipulada.



A próxima atividade vai exigir um uso mais sofisticado de condicionais do que as atividades que resolvemos até agora.

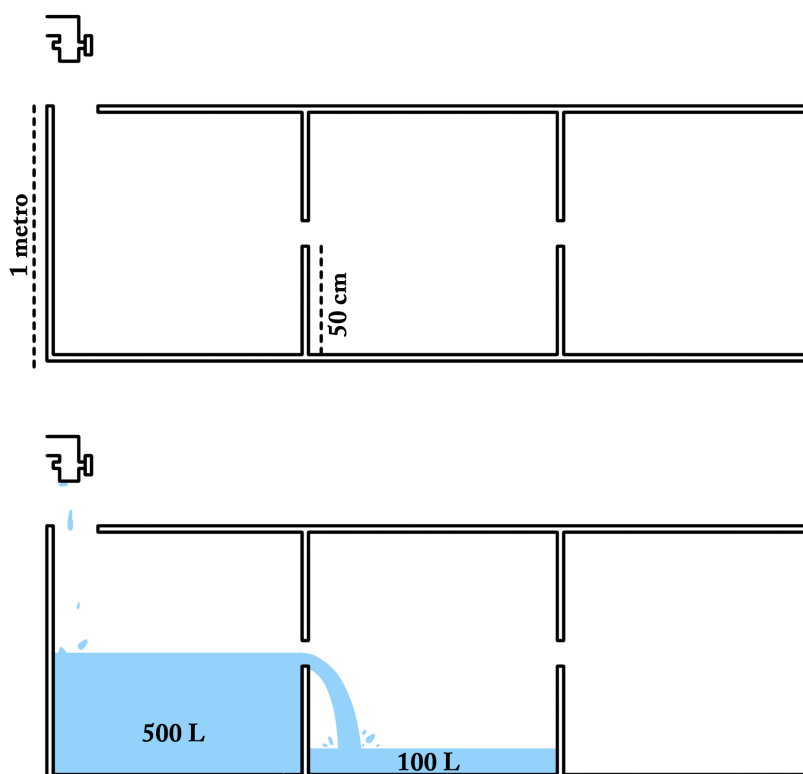


Atividade 6

Cubos conectados

Uma empresa que produz detergente transporta o produto bruto em cubos metálicos com lados iguais a 1 metro. Para facilitar o enchimento destes cubos, eles possuem uma abertura pequena, como mostrado na figura abaixo, que permite conectar três cubos durante o enchimento. Assim, à medida que o líquido é despejado, ao atingir a altura da abertura, o líquido começa a escoar para o segundo cubo. Se houver líquido suficiente, o terceiro cubo também poderá receber uma parte.

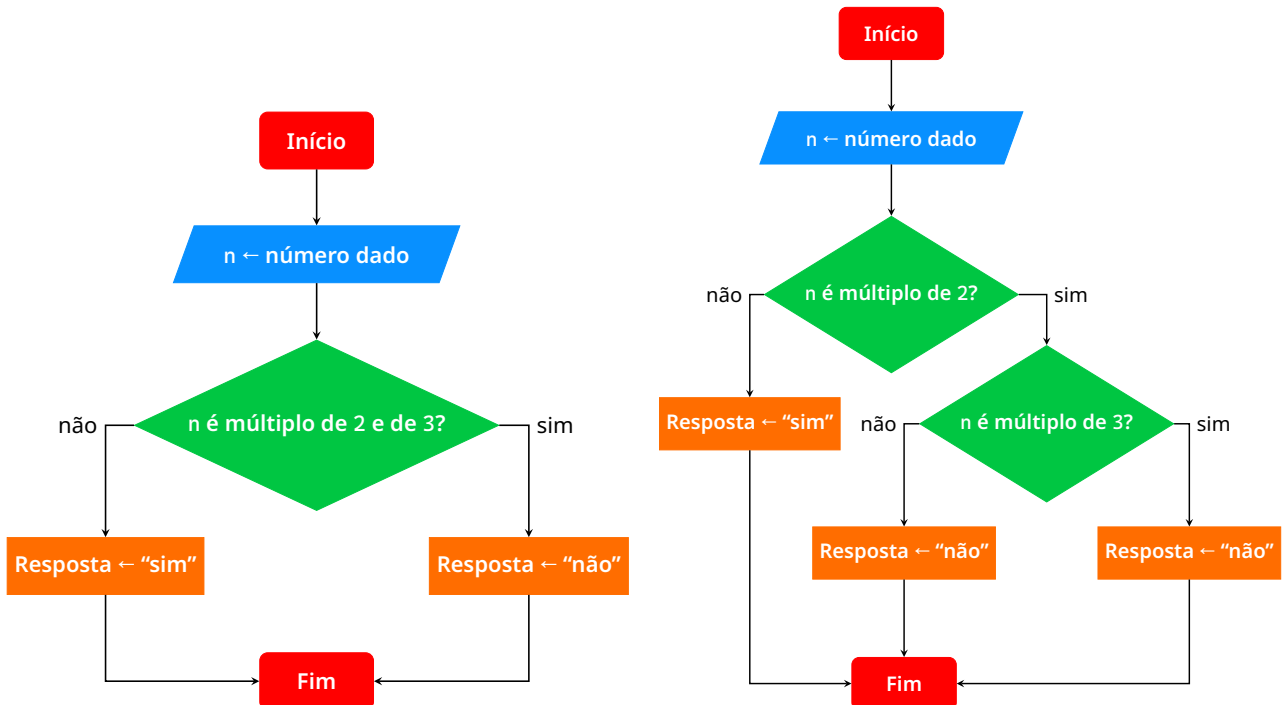
Por exemplo, se forem injetados 600 litros de detergente, os cubos ficarão como mostrado a seguir, com 500 litros no primeiro, 100 litros no segundo e 0 no terceiro.



- Quantos litros haverá em cada cubo se forem injetados 900 litros de detergente?
- Quantos litros haverá em cada cubo se forem injetados 1250 litros de detergente?
- Quantos litros haverá em cada cubo se forem injetados 1800 litros de detergente?
- Quantos litros haverá em cada cubo se forem injetados 2400 litros de detergente?
- A empresa quer que você crie um algoritmo que permita saber qual é o volume de detergente em cada um dos três cubos quando uma quantidade N (em litros) é despejada no conjunto. Apresente a sua solução de maneira esquemática, isto é, sem usar muito texto.



Leia com atenção os dois fluxogramas abaixo. Ambos foram criados para determinar se um número dado é múltiplo de 6 a partir do seguinte resultado: um número é múltiplo de 6 se for múltiplo de 2 e de 3.



Para refletir

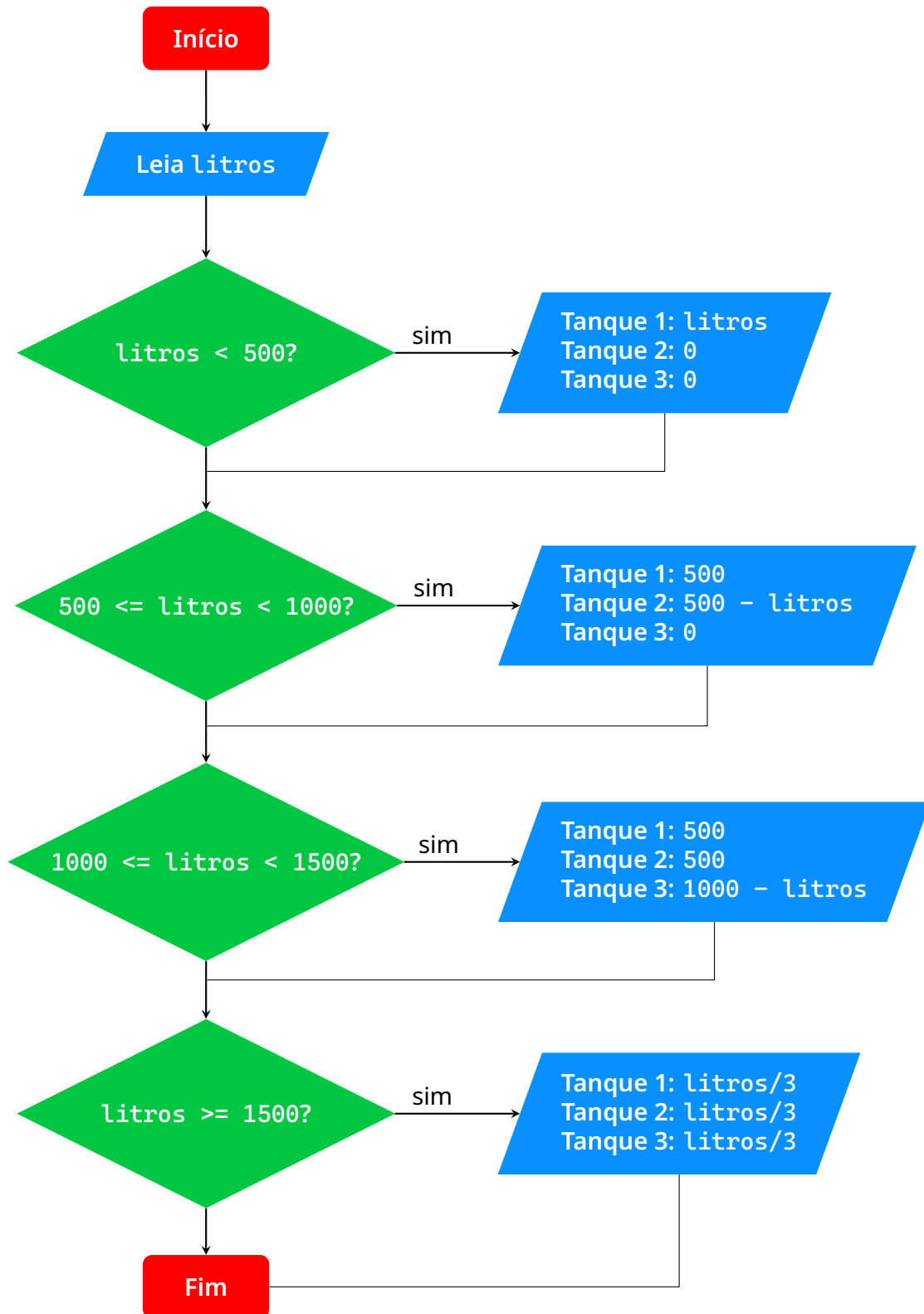
Os dois fluxogramas são visualmente diferentes, mas o que você pode dizer sobre os resultados que cada um deles produz para diferentes valores de n ?

Veja que no fluxograma da esquerda, temos uma única condição e ela testa, de uma só vez, se n é múltiplo de 2 e se n é múltiplo de 3. No fluxograma da direita temos duas condições e a segunda está “dentro” da primeira, sendo acionada apenas se a primeira for verdadeira.

Apesar de estruturalmente diferentes, ambos produzem os mesmos resultados para qualquer valor de n . Como ambas estão corretas, a escolha por uma ou outra solução deve ser feita por outros critérios, como a clareza para o leitor, os recursos disponíveis na linguagem de programação que você esteja usando ou até mesmo considerações sobre a eficiência de cada algoritmo em termos de velocidade de processamento computacional (mas essa discussão é muito mais avançada do que os nossos objetivos no momento).

Agora, vejamos duas soluções para a atividade [Cubos Conectados](#) representadas como um fluxograma e como um algoritmo em Portugol.





Nesse fluxograma, as quatro condições são sempre analisadas, não importando se alguma delas já foi satisfeita e produziu a resposta corretamente. O algoritmo em Portugol que reproduz esse comportamento esta colocado a seguir:

```

1 programa {
2   funcao inicio() {
3     real litros
4     leia(litros)
  
```



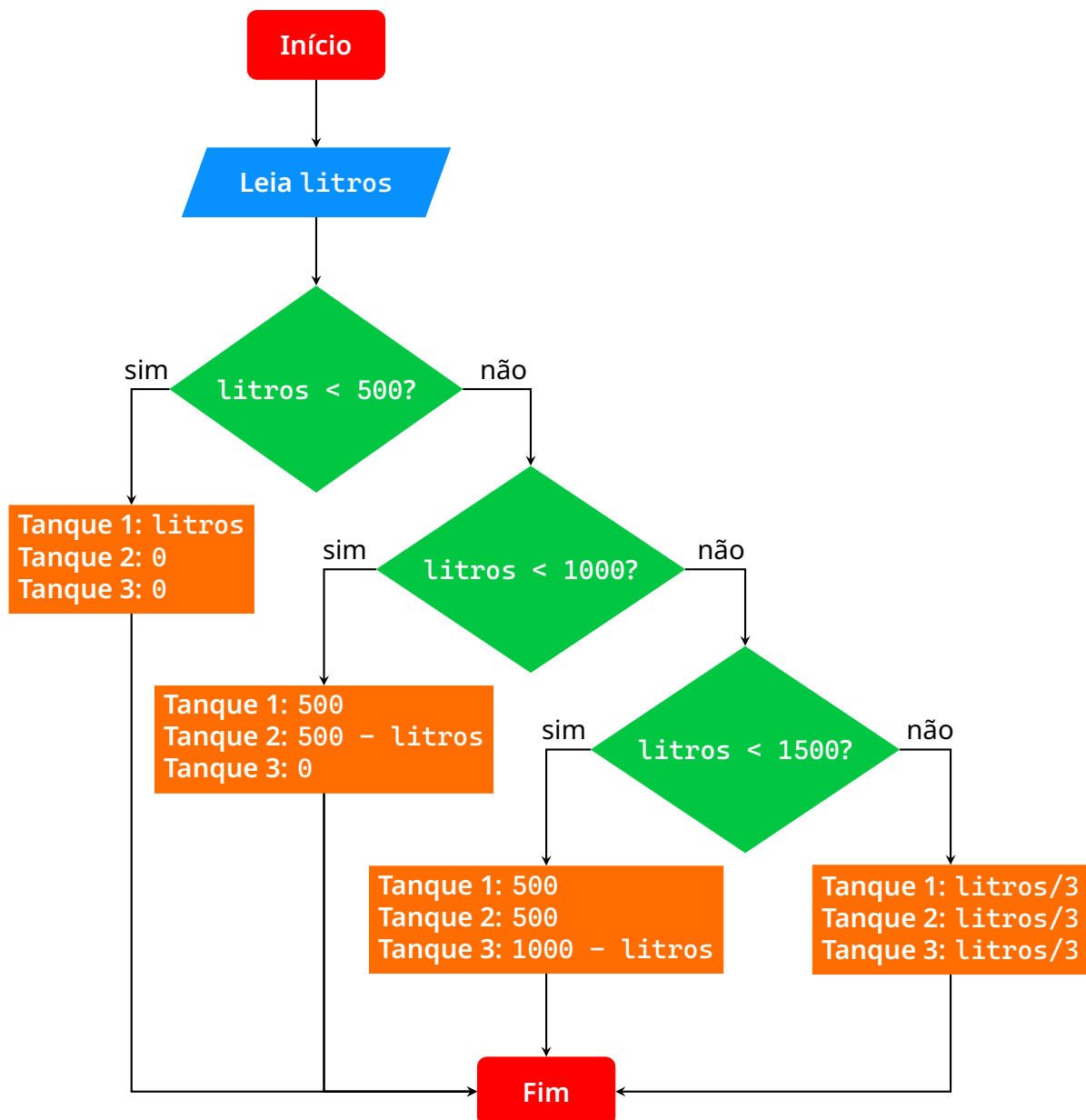
```

5  se (litros<500) {
6      escreva(litros+", 0, 0")
7  }
8  se (litros>=500 e litros<1000) {
9      escreva("500, " + (litros-500) + ", 0")
10 }
11 se (litros>=1000 e litros<1500) {
12     escreva("500, 500, " + (litros-1000))
13 }
14 se (litros>=1500) {
15     escreva((litros/3) + ", " + (litros/3) + ", " + (litros/3))
16 }
17 }
18 }

```

Os quatro comandos **se** que compõem o código são independentes um do outro e o algoritmo, ao ser executado, checará as quatro condições.

Uma outra opção é mostrada no fluxograma a seguir:



Note que nesse fluxograma, as condições estão encadeadas: se o total de litros colocados for menor do que 500, o fluxo do algoritmo passa pelo bloquinho azul mais à esquerda e depois já vai para o fim, sem sequer considerar as demais condições. Dessa forma, o algoritmo evita checagens desnecessárias.

O algoritmo em Portugol que reproduz esse comportamento está colocado a seguir:

```
1 programa {
2   funcao inicio() {
3     real litros
4     leia(litros)
5     se (litros<500) {
6       escreva(litros+", 0, 0")
7     }
8     senao {
9       se (litros<1000) {
10        escreva("500, " + (litros-500) + ", 0")
11      }
12      senao {
13        se (litros<1500) {
14          escreva("500, 500, " + (litros-1000))
15        }
16        senao {
17          escreva(litros/3+" "+litros/3+" "+litros/3)
18        }
19      }
20    }
21  }
22 }
```

Veja que agora todo comando **se** é seguido por um **senao**. Isso faz com que o algoritmo economize algumas checagens, pois os comandos que estão dentro de um **senao** só são executados se a condição do **se** ao qual ele está ligado não for satisfeita.

Ainda há outras formas corretas e diferentes de resolver esse problema, mas o importante neste momento é ter clareza de que as soluções acima são equivalentes, apesar de estruturadas de forma diferente.



4 REPETIÇÕES

Repetir muitas vezes uma mesma ação é algo considerado cansativo para a maioria das pessoas. Porém, para muitas tarefas isso é necessário, e esse é um dos motivos que fazem dos computadores uma ferramenta tão útil: eles podem repetir comandos sem se cansar, sem perder a motivação ou a atenção e sem se distrair a ponto de cometer erros.

Historicamente, essa foi a motivação de diversos cientistas que, a partir do século 17, começaram a propor máquinas que fossem capazes de realizar cálculos aritméticos e que podem ser consideradas as avós dos computadores modernos.

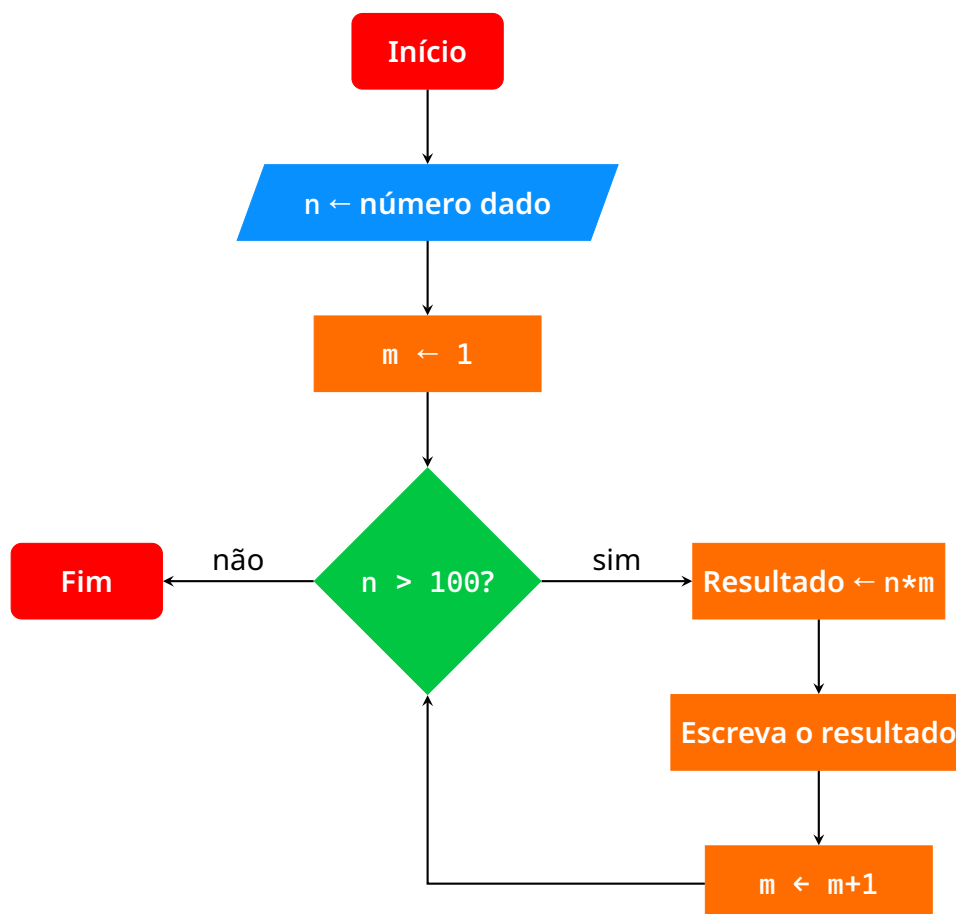


Você Sabia?

As primeiras máquinas concebidas para realizar cálculos eram puramente mecânicas, ou seja, baseadas em engrenagens e encaixes físicos. Pesquise na internet pelas máquinas desenvolvidas pelos matemáticos Blaise Pascal e Gottfried Leibniz.

Nas próximas atividades, vamos lidar com problemas que envolvem muitas repetições e que ainda são muito relevantes na atualidade. Por isso, vamos aprender como criar algoritmos que realizem essas repetições por nós.

Vamos começar com um algoritmo simples que contém um ciclo de repetição representado na forma de um fluxograma:



Note que as três ações mostradas na parte de baixo à direita do fluxograma são repetidas diversas vezes, enquanto a condição $m < 10$ for verdadeira.



A cada repetição, a variável m tem o seu conteúdo aumentado em uma unidade, o que faz com que, em algum momento, o seu valor seja maior ou igual a 10, violando a condição e fazendo com que o algoritmo saia do ciclo para o seu fim.



Para refletir

O que este algoritmo escreverá se o número dado for 8?

Em Portugol, ciclos são realizados pelo comando **enquanto**: quando o computador chega neste comando, ele entende que deve repetir os comandos dentro das chaves seguintes enquanto a condição colocada ($m < 10$) for verdadeira.

```
1  programa {
2  funcao inicio() {
3      inteiro n, m, resultado
4      leia(n)
5      m=1
6      enquanto (m<10) {
7          resultado = m*n
8          escreva(resultado + "\n")
9          m=m+1
10     }
11 }
12 }
```

Esse é um exemplo simples de uma estrutura de repetição, que poderia ser feita por uma pessoa sem grande esforço. Mas ela transmite muito bem a ideia de repetir alguma ação enquanto uma condição for verdadeira.

Nas atividades a seguir, vamos propor algumas tarefas repetitivas que são muito trabalhosas quando feitas manualmente, por isso é ainda mais relevante utilizar um computador para fazê-las.

Explorando

Repetir e repetir

Atividade 7

É primo ou não?

Você deve se lembrar que um número é chamado primo se tiver apenas dois divisores inteiros positivos: 1 e ele mesmo. Por exemplo, o número 9 não é primo, pois ele também é divisível por 3 enquanto que 13 é primo, pois só é divisível por 1 e por ele mesmo.





Até hoje, não existem métodos realmente rápidos para identificar se um número é primo ou não. Basicamente, todos os métodos existentes baseiam-se em testar se o número dado deixa resto zero quando é dividido pelos números menores do que ele. Imagina a quantidade de repetições necessárias para verificar se o número 4.000.037 é primo!

- Descreva um algoritmo que verifica se um número é primo ou não.
- Em conjunto com toda a turma, escreva um dos algoritmos criados no item anterior em Português.
- Utilize esse algoritmo para verificar se o número 4.000.037 é primo.
Você deve ter notado que o algoritmo responde quase que imediatamente se o número dado é primo ou não, mesmo sendo um número grande.
- Se você conseguisse verificar um divisor a cada 1 segundo, estime aproximadamente quanto tempo você teria levado para verificar se 4.000.037 é primo.



Para refletir

Embora muitas checagens precisem ser feitas, não é necessário dividir um número dado por todos os números naturais menores do que ele para verificar se ele é primo. Várias melhorias podem ser feitas nesse processo para torná-lo mais eficiente. Discuta com a turma essas melhorias.

Atividade 8

A sequência de Collatz

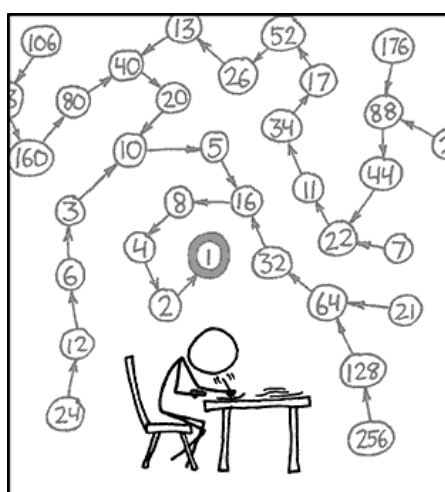
A sequência de Collatz é uma sequência numérica formada por números inteiros que é construída a partir de um valor inicial, que podemos chamar de a_1 , por meio da seguinte regra: o próximo termo da sequência é igual à metade do anterior, se o anterior for par, e igual ao triplo do anterior mais um, se o anterior for ímpar. Matematicamente, podemos descrever essa regra de formação dessa maneira:

$$a_n = \begin{cases} a_{n-1}/2, & \text{se } a_{n-1} \text{ é par} \\ 3a_{n-1} + 1, & \text{se } a_{n-1} \text{ é ímpar} \end{cases}$$



Por exemplo, se tomarmos, temos a seguinte sequência: 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, Note que a sequência entra em um ciclo depois que chega em 1 pela primeira vez. Por isso, dizemos que ela termina ao chegar em 1, ou seja, a sequência de Collatz com $a_1 = 5$ tem 6 termos: 5, 16, 8, 4, 2, 1.

- Escreva a sequência de Collatz para $a_1 = 12$.
- Escreva a sequência de Collatz para $a_1 = 17$.
- Escolha um número entre 5 e 20 e obtenha a sequência de Collatz iniciada por esse número.
- Escreva em Portugol um algoritmo que obtenha a sequência de Collatz a partir de um valor dado para a_1 .
- Com auxílio do seu algoritmo, determine a sequência de Collatz para $a_1 = 26$.



THE COLLATZ CONJECTURE STATES THAT IF YOU PICK A NUMBER, AND IF IT'S EVEN DIVIDE IT BY TWO AND IF IT'S ODD MULTIPLY IT BY THREE AND ADD ONE, AND YOU REPEAT THIS PROCEDURE LONG ENOUGH, EVENTUALLY YOUR FRIENDS WILL STOP CALLING TO SEE IF YOU WANT TO HANG OUT.

Figura 3: Fonte: <https://xkcd.com/710>. O texto diz: A conjectura de Collatz afirma que se você escolher um número, sendo par dividi-lo por 2, sendo ímpar multiplicá-lo por 3 e somar 1, e se você repetir isso por tempo suficiente, seus amigos vão parar de te ligar pra saber se você quer encontrá-los.

O comportamento da sequência de Collatz pode ser surpreendente. No caso do item **e)** acima, a sequência tem apenas 11 termos, mas se usarmos $a_1 = 27$ a sequência precisa de 113 termos para terminar! Dica: você pode criar uma nova variável para contar os termos da sequência.

É claro que repetir esse processo tantas vezes pode ser bem entediante, mas esse é o tipo de tarefa que um computador pode fazer sem dificuldade e muito mais rapidamente. Com o algoritmo que você criou em mãos, é mais simples testar casos e eventualmente levantar conjecturas sobre essa sequência: é possível prever valores de que geram sequências bem curtas? Há alguma família de números que, se atingida, encaminha a sequência para o seu final? Será que qualquer que seja o valor de a_1 a sequência sempre termina?

Um detalhe interessante é que até hoje os matemáticos não conseguiram responder satisfatoriamente a essa última pergunta! Com o auxílio de computadores, muitos valores de a_1 já foram testados e a sequência gerada a partir deles sempre chega, cedo ou tarde, em 1. Mas ninguém conseguiu provar que isso é verdadeiro para qualquer número inteiro positivo desde que o problema foi proposto (há mais de 80 anos).

Processos que envolvem muitas repetições são bons exemplos de como os computadores podem ser usados como ferramenta: podemos programá-los para executar essa parte da tarefa e focarmos nas partes que envolvem criatividade e o reconhecimento de padrões. No caso da sequência de Collatz, poder analisar e comparar várias sequências rapidamente nos permite levantar conjecturas, testá-las e ajustá-las de acordo com as observações.

Para compreendermos bem o uso de repetição, vamos analisar o algoritmo abaixo.

Ele escreve os termos de uma sequência cujos dois primeiros termos valem 1 (veja que as variáveis `a1` e `a2` começam recebendo esses valores). A variável `n`, que é lida no início do algoritmo determina a quantidade de termos que serão escritos. Veja que os dois primeiros são escritos antes do comando `enquanto`, por isso a variável `i` recebe o valor 3 (quando o `enquanto` começar, já estaremos escrevendo o terceiro termo).

```

1  programa {
2  funcao inicio() {
3      inteiro a1,a2,an,penultimo,ultimo,proximo,n,i
4      a1=1
5      a2=1
6      escreva(a1+" "+ a2+" ")
7      ultimo=a2
8      penultimo=a1
9      leia(n)
10     i=3
11     enquanto (i<=n) {
12         proximo=penultimo+ultimo
13         escreva(proximo+" ")
14         i=i+1
15         penultimo=ultimo
16         ultimo=proximo
17     }
18 }
19 }
```



Para refletir

Quais valores seriam escritos considerando $n = 6$? Você conhece essa sequência numérica?

O aspecto que é importante compreender neste algoritmo é o uso das variáveis `i` e `proximo`.

Note que a variável `i` começa com o valor 3 e ela aparece na condição que determina quantas vezes o comando `enquanto` vai ser repetido. Ao final dos comandos que são executados dentro do `enquanto`, o valor de `i` é incrementado em uma unidade, até que ela fica maior do que `n` e o `enquanto` para de ser repetido. O papel dessa variável é servir como um contador de repetições.

Já as variáveis `ultimo`, `penultimo` e `proximo` têm uma função diferente. Para entendê-las, é necessário analisar os comandos das linhas 7 e 8 e depois das linhas 12, 15 e 16.

Nas linhas 7 e 8, as variáveis `ultimo` e `penultimo` recebem o valor das variáveis `a1` e `a2`. Note que a partir desse ponto, `a1` e `a2` não são mais utilizadas. Elas foram usadas para registrar o início da sequência numérica e depois os termos seguintes serão criados usando as variáveis `ultimo`, `penultimo` e `proximo`. Veja que na linha 7, a variável `ultimo` recebe o valor do último



termo conhecido, nesse caso o **a2**, e na linha 8, a variável **penultimo** recebe o valor do penúltimo termo conhecido, nesse caso o **a1**.

Na linha 12, a variável **proximo** recebe o valor de **penultimo+ultimo** e depois o seu conteúdo é escrito, ou seja, **proximo** é o próximo termo da sequência.

Por fim, os comandos das linhas 15 e 16 estão fazendo o trabalho de “avançar na sequência”. Nessas linhas, variáveis **penultimo** e **ultimo** deixam de ser o primeiro e segundo termos e passam a ser os próximos: **penultimo** recebe o valor do **ultimo** e **ultimo** recebe o valor de **proximo**, como se estivéssemos “andando” na sequência. Dessa forma, na próxima repetição, **penultimo** e **ultimo** estarão armazenando os dois últimos valores calculados e **proximo** será a soma deles, ou seja, o termo seguinte.

Você pode adaptar esse algoritmo, e o que aprendeu com as atividades anteriores, para obter outras sequências numéricas, como progressões aritméticas, progressões geométricas e os números triangulares. Pesquise na internet sobre essas sequências e tente construir algoritmos em Portugol para cada uma delas.



5 PRATICANDO TUDO QUE FOI ESTUDADO

Nesta seção, vamos praticar tudo que foi estudado até este momento.

Todas as atividades propostas envolvem dois aspectos, o matemático e o computacional. Isso significa que conhecimentos matemáticos serão necessários para compreender e resolver os problemas e você também terá que pensar computacionalmente sobre o processo de resolução para transformar a sua resolução em um algoritmo.

Apesar das atividades não pedirem explicitamente, você pode criar todos os seus algoritmos em Portugol e testá-los usando as respostas que você obteve ao longo da atividade.

Praticando

Tudo o que você aprendeu

Atividade 9

Imposto de renda retido na fonte

O imposto de renda é um dos principais impostos no Brasil. Uma das formas em que esse imposto é cobrado ocorre na folha de pagamento, ou seja, quando um funcionário com carteira de trabalho assinada recebe o seu salário. Essa forma de cobrança é chamada de imposto de renda retido na fonte.

O cálculo do valor a ser descontado do salário (e imediatamente repassado ao governo federal) é feito de forma que ele seja percentualmente maior à medida que o salário aumenta. Em 2020 eram adotados 5 intervalos de salário bruto (isto é, ainda sem o desconto do imposto). Para cada um desses intervalos, o cálculo é feito da seguinte maneira: calcula-se a porcentagem indicada em “alíquota” do salário bruto e, do resultado, subtrai-se a “parcela a deduzir”. O resultado obtido é o valor que será descontado do salário no momento do pagamento.

| Base de cálculo (R\$) | Alíquota (%) | Parcela a deduzir do IR (R\$) |
|--------------------------|--------------|-------------------------------|
| Até 1.903,98 | — | — |
| De 1.903,99 até 2.826,65 | 7,5 | 142,80 |
| De 2.826,66 até 3.751,05 | 15,5 | 354,80 |
| 3.751,06 até 4.664,68 | 22,5 | 636,13 |
| Acima de 4.664,68 | 27,5 | 869,36 |

Tabela 1: Fonte: [Receita Federal](#)

Por exemplo, para um salário de R\$ 2.000,00, começamos identificando que ele se encontra na segunda faixa. Logo, devemos calcular 7,5% de 2000 e depois subtrair 142,80 do resultado. Portanto, $0,075 \times 2000 - 142,80 = 7,20$ é o valor que será descontado do salário para pagar o imposto de renda desse trabalhador.

- Quanto será descontado de uma pessoa que tenha salário bruto igual a R\$ 1.500,00? E igual a R\$ 3.000,00? E igual a R\$ 6.000,00?
- Crie um algoritmo em Portugol que calcule o valor do imposto de renda a ser retido na fonte para um dado salário bruto.



Xadrez

No tabuleiro de xadrez, a casa na linha 1, coluna 1 (canto superior esquerdo) é sempre branca e as cores das casas se alternam entre branca e preta. Dessa forma, como o tabuleiro tradicional tem oito linhas e oito colunas, a casa na linha 8, coluna 8 (canto inferior direito) será também branca. Neste problema, entretanto, queremos saber a cor da casa no canto inferior direito de um tabuleiro com dimensões quaisquer: L linhas e C colunas. No exemplo da figura, para $L = 6$ e $C = 9$, a casa no canto inferior direito será preta.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1 | Preta | Branca | Preta | Branca | Preta | Branca | Preta | Branca | Preta |
| 2 | Branca | Preta | Branca | Preta | Branca | Preta | Branca | Preta | Branca |
| 3 | Preta | Branca | Preta | Branca | Preta | Branca | Preta | Branca | Preta |
| 4 | Branca | Preta | Branca | Preta | Branca | Preta | Branca | Preta | Branca |
| 5 | Preta | Branca | Preta | Branca | Preta | Branca | Preta | Branca | Preta |
| 6 | Branca | Preta | Branca | Preta | Branca | Preta | Branca | Preta | Branca |

- Qual seria a cor da casa no canto inferior direito se o tabuleiro tiver 7 colunas e 4 linhas? E se tiver 10 colunas e 8 linhas? E se tiver 23 colunas e 40 linhas?
- Descreva como descobrir a cor da casa no canto inferior direito de um tabuleiro como esse, sabendo a quantidade de linhas e colunas que o compõe.
- Escreva um algoritmo em Portugol que implemente o processo que você descreveu na questão anterior.

MMC

Existem vários métodos para o cálculo do mínimo múltiplo comum (MMC) entre dois números inteiros. Você deve ter aprendido como fazer isso ainda no Ensino Fundamental e, de vez em quando, ainda deve utilizar esse procedimento para resolver algumas questões de matemática.

- Descreva como você procede para obter o mínimo múltiplo comum entre dois números dados. Se quiser, comece obtendo o mínimo múltiplo comum para os números 12 e 18 e depois tente descrever o método que você utilizou de forma genérica, isto é, para dois números quaisquer a e b .

Embora não seja o método mais eficiente, vamos implementar o método da lista. Ele consiste em listar os múltiplos do primeiro número (começando por ele mesmo) até encontrar um múltiplo que seja divisível pelo segundo número.

Exemplo: para o caso 12 e 18, vamos listar os múltiplos de 12. Primeiro, o próprio 12, que não é divisível por 18. Depois 24 ($12 \cdot 2$), que não é divisível por 18. Depois 36 ($12 \cdot 3$), que é divisível por 18 e, portanto, é o mínimo múltiplo comum entre esse 12 e 18.

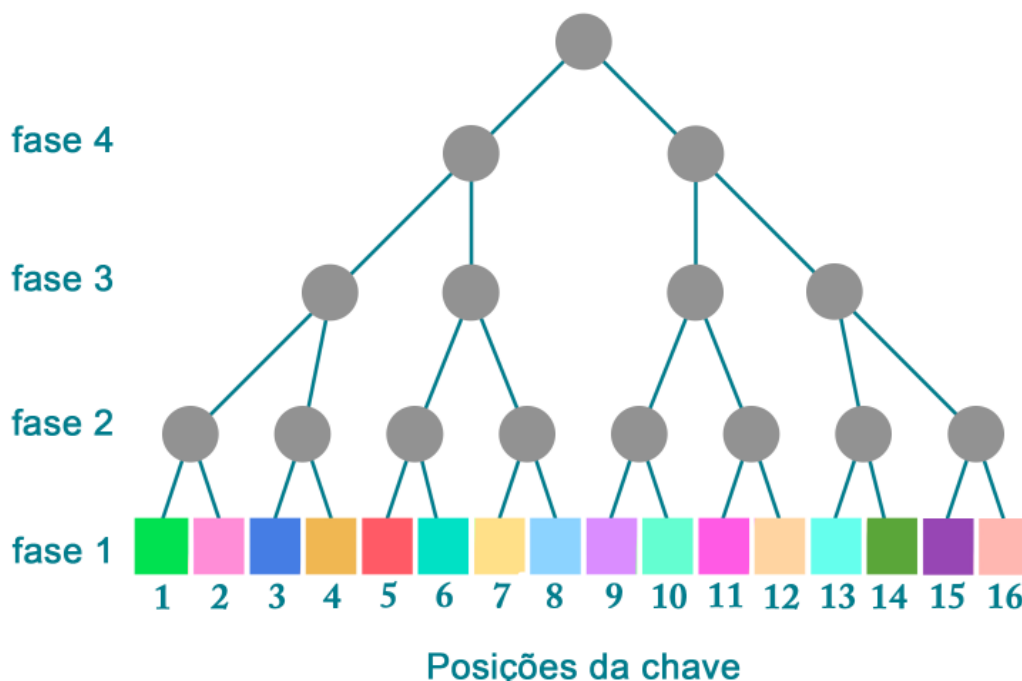


- b) Escreva um algoritmo em Portugol que obtenha o mínimo múltiplo comum entre dois números naturais dados, usando o método da lista.

Atividade 12

Campeonato

Um torneio esportivo mundial é organizado no tradicional formato de chaves: dois competidores se enfrentam e quem vence passa para a próxima etapa, até que os dois que venceram todas as partidas se enfrentem na final. Nesta edição, o torneio vai contar com 16 competidores e os confrontos ocorrerão como mostrado abaixo.



O grande adversário do Brasil neste torneio é os Estados Unidos e, por isso, todos querem saber quando os competidores desses dois países poderão vir a se enfrentar. Isso será determinado um dia antes do início do torneio, quando será sorteada uma bolinha com um número de 1 a 16 para cada país, indicado a posição em que eles serão alocados nas chaves.

- a) Escreva um algoritmo em Portugol que, dadas as duas posições do Brasil e Estados Unidos nas chaves, determina em qual fase do torneio os países poderão se enfrentar.



Para refletir

Você conseguiria modificar o seu algoritmo de modo que ele possa ser usado para outras quantidades de competidores?

Observação: você pode assumir que essas quantidades sempre serão uma potência de 2.



REFERÊNCIAS

- Brasil. (2018). *Base Nacional Comum Curricular* [Disponível em: <http://basenacionalcomum.mec.gov.br>]. Ministério da Educação. Brasília, Brasil.
- Disessa, A. A. (2018). A Computation Literacy and “The Big Picture” Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, 20(1), 3–31.
- Doolittle, C. L. (1910). Halley’s Comet. *The Popular Science Monthly*.
- Esteves, A., Noschang, L., Raabe, A., & Filho, A. (2019). Portugol Studio: Em direção a uma comunidade aberta para pesquisa sobre o aprendizado de programação. *Anais do XXVII Workshop sobre Educação em Computação*, 513–522.
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational Thinking Is More about Thinking than Computing. *Journal for STEM Education Reserach*, 1–18.
- Noschang, L., Pelz, F., de Jesus, E., & Raabe, A. (2014). Portugol Studio: IDE para Iniciantes em Programação. *Anais do XXII Workshop sobre Educação em Computação*, 1–10.
- Raabe, A., Zorzo, A. F., & Blikstein, P. (2020). *Computação na Educação Básica: Fundamentos e Experiências*. Penso Editora.
- Reis, S. R. d., Barichello, L., & Mathias, C. V. (2021). Novos conteúdos e novas habilidades para a área de Matemática e suas Tecnologias. *Revista Internacional De Pesquisa Em Educação Matemática*, 11(1), 37–58. <https://doi.org/https://doi.org/10.37001/ripem.v11i1.2539>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158.



Coleção Livro Aberto de Matemática

O esforço para produzir livros didáticos de matemática com licença aberta começou em 2016, com a elaboração do material de Frações para o Ensino Fundamental I. Desde então, novos elaboradores acreditaram e juntaram-se ao projeto para alcançarmos novos níveis e novos livros.

Hoje, possuímos diversos capítulos escritos e vários livros em produção. Tudo isso a partir de um trabalho colaborativo envolvendo matemáticos, professores universitário e professores da Educação Básica.

Um princípio fundamental desta iniciativa é que sua produção configure uma proposta pedagógica ancorada e acompanhada por pesquisa científica em Ensino de Matemática.

O projeto tem também compromisso com a formação e o desenvolvimento profissional de professores. Em particular, pela composição característica da equipe, destaca-se o entendimento do potencial do projeto para enfrentar um reconhecido desafio: estreitar o diálogo entre a realidade e as demandas próprias da prática docente e a formação acadêmica do professor.

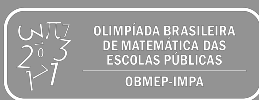
O Livro Aberto de Matemática é um projeto do IMPA, desenvolvido em suas etapas iniciais pela Associação Livro Aberto com financiamento da Fundação Itaú Social e apoio da UNIRIO e da UFRJ.

Realização

impa



Instituto de
Matemática
Pura e Aplicada



Somando novos talentos para o Brasil

Patrocínio



Itaú Social

