

Pensamento Computacional

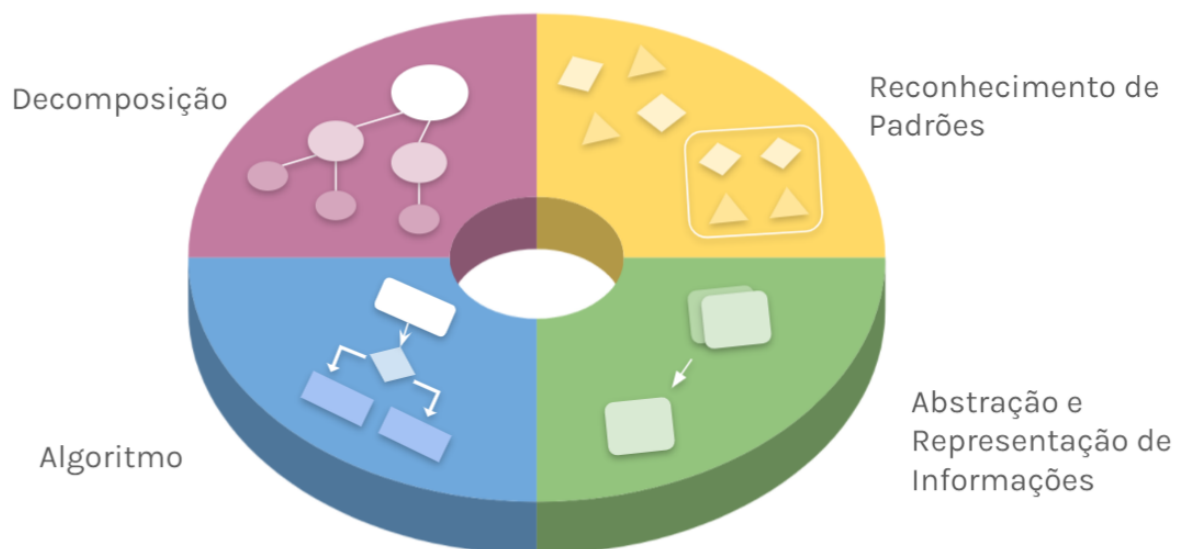
Introdução

Aprender qualquer coisa nova pode ser confuso, principalmente quando nos desafia a pensar de uma forma que não estamos acostumados no dia-a-dia. Esse é o caso da Programação!

Assim como uma língua estrangeira, aprender uma linguagem de programação faz nosso cérebro enxergar as coisas de forma diferente. É uma nova gramática, uma nova semântica e um novo raciocínio. E só aprendemos isso com a prática.

Para facilitar o processo de aprendizado, podemos utilizar os 4 pilares do pensamento computacional para auxiliar na resolução de problemas.

Quatro Pilares do Pensamento Computacional



01. Decomposição

O primeiro pilar é a **decomposição**. Ou como gostamos de dizer na área da computação: **dividir para conquistar**. Se temos um problema grande, podemos dividi-lo em subproblemas mais simples. Isso nos permite dar atenção a um pedaço do desafio por vez e, ao resolvê-lo, temos também uma solução parcial para o problema inteiro. No fim, basta juntar cada subsolução para resolver o problema como um todo.

02. Reconhecimento de Padrões

O segundo é o **reconhecimento de padrões**. A habilidade de reconhecer padrões é algo muito frequente no nosso cotidiano e utilizamos disso sempre que estamos aprendendo algo novo. Dentro do pensamento computacional, utilizamos-a para identificar, nas variações do problema, o que há em comum entre elas.

Por exemplo, a máquina de lavar roupa é uma única solução para um mesmo problema: roupas sujas. Para lavar qualquer roupa ela utiliza sabão e água (e às vezes amaciante). Mas algumas peças possuem necessidades diferentes: um edredom precisa de uma lavagem mais pesada enquanto as roupas de bebê precisam de uma lavagem mais delicada. Apesar disso, as duas lavagens têm muita coisa em comum, esse é o padrão entre elas. E é isso que permite usarmos apenas uma máquina para lavar qualquer tipo de roupa, ao invés de ter uma máquina para cada característica.

É a mesma coisa dentro da programação. Um programa que realiza cálculos numéricos pode precisar diferenciar números naturais de inteiros, mas deve aproveitar o máximo possível da solução ao generalizá-la para atender qualquer variante do problema. É sobre encontrar o que cada caso tem em comum e o que podemos fazer para solucioná-los.

03. Abstração e Representação de Informações

O terceiro pilar é a **Abstração e Representação de Informações**. Esse pilar diz respeito a identificar as informações importantes do problema e criar um "modelo" que represente o mesmo.

Por exemplo, a tarefa de cadastrar um livro em um sistema de biblioteca. Há diversas informações importantes sobre o livro: código ISBN, nome, autores, edição, entre outros. Mas há informações que não são tão importantes, como a cor, a altura dele ou o seu conteúdo na íntegra.

Em seguida, podemos montar uma representação simplificada do livro. Isto é, não é o livro integralmente, mas é tudo que um livro dentro do sistema precisa ter.

Assim, para cada problema devemos separar os dados essenciais e criar uma representação simplificada da entrada do problema, o que nos ajuda a montar a solução.

04. Algoritmo

Por último, temos o **Algoritmo**. Apesar de parecer um nome complicado, algoritmos não são nada mais do que uma sequência de passos que precisa ser realizada.

É como uma receita de cozinha. Os ingredientes podem ser considerados nossa entrada (o que utilizamos para fazer o prato), o modo de preparo é o nosso algoritmo (passo-a-passo) e o prato final é o resultado.

Dessa forma, podemos dividir todo problema em entrada, processamento e saída. A entrada é o que o programa precisa saber, o processamento é a sequência de passos para resolver o problema (algoritmo) e a saída é o resultado do programa.

É nesse pilar que realmente determinamos o que é necessário para resolver o problema.

Exemplos

Para treinar a aplicação desses pilares, vamos ver dois exemplos. O primeiro deles já foi apresentado no vídeo que acompanha esse material:

01. Quantas vogais há nesta palavra?

Queremos ser capazes de, recebendo qualquer palavra como entrada, contar quantas vogais há nela. A proposta parece simples, mas vamos aplicar os 4 pilares para ver como podemos traduzi-la com facilidade para que um computador a entenda.

Vamos começar pela **decomposição**. Pode parecer um problema simples, mas podemos dividi-lo em várias partes. Então, conseguimos dividir em 3 subproblemas:

1. Ler a palavra, letra a letra;
2. Identificar se a letra é vogal ou consoante;
3. Contar as vogais.

Agora que temos nosso problema subdividido, podemos aplicar o segundo pilar, **reconhecimento de padrão**.

Nossa solução tem que atender a qualquer palavra. Seja uma pequenina como “pão” ou uma grande como “anticonstitucionalíssimamente”. Ou seja, não podemos ensinar nosso programa a ler palavras de no máximo 3 letras nem a contar no máximo 5 vogais.

Precisamos encontrar o que há em comum com todas as palavras: são compostas por letras, sem números ou espaços no meio e que acaba com um espaço em branco ou um sinal de pontuação.

Para o terceiro pilar, de **abstração**, há muitas informações que podem estar relacionadas com esse nosso problema. O número de sílabas de cada palavra, seus fonemas ou até a quantidade de caracteres. Mas nenhuma dessas informações são necessárias para nossa solução.

As únicas informações que precisam ser fornecidas ao programa são a palavra a ser contada e quais são as vogais do nosso alfabeto.

Por último, temos nosso **algoritmo**. Dado o discutido sobre os pilares anteriores, podemos montar nosso algoritmo dessa forma:

1. **Primeiro passo:** Receber a palavra a ser contada;
2. **Segundo passo:** Começando pela esquerda, pegar a primeira letra;
 - a. A cada letra, comparar com A | E | I | O | U.
 - b. Caso seja alguma dessas vogais, somamos 1 ao nosso contador de vogais.
 - c. Seguir para a próxima letra.
3. **Terceiro passo:** Caso não tenha mais letras, o valor resultante em nosso contador é o número de vogais.

A descrição acima, apesar de estar escrita em português, é facilmente traduzida para qualquer linguagem de programação que possa resolver este problema.

02. Venda de Ingressos em um Congresso

Agora vejamos um outro exemplo. Imagine que o Congresso Nacional de Matemática Aplicada e Computacional (CNMAC) está precisando de um novo sistema para a venda das inscrições e para isso contratou a sua empresa para desenvolver um software. O programa deve funcionar da seguinte maneira:

- O usuário informa de que maneira deseja participar do evento, podendo escolher entre ouvinte ou apresentador. Para ambos os casos, o usuário poderá se inscrever como Profissional, Estudante de Pós-Graduação ou Estudante de Graduação e Professor de Ensino Básico.
- O software retorna a taxa de inscrição de acordo com o perfil desejado para a compra. Para isso é preciso considerar também a categoria selecionada, entre usuário associado ou não-associado da Instituição responsável pelo evento.

Esse problema pode parecer complexo, mas aplicando os 4 pilares apresentados anteriormente, podemos simplificá-lo um pouco.

Primeiramente, precisamos **decompor** esse problema. Observe que podemos dividi-lo nas seguintes “tarefas” que o software precisa realizar:

1. Leitura dos dados informados pelo usuário
2. Identificação do tipo de inscrição a ser realizada de acordo o perfil selecionado
3. Consulta na base de dados do valor da taxa para esse tipo de inscrição
4. Apresentar o valor da inscrição para o usuário
5. Realizar a venda da inscrição

Em seguida, vamos trabalhar o **reconhecimento de padrões**. Temos 12 casos possíveis de tipo de inscrição a ser escolhida pelo usuário. Visto que temos:

2 (ouvinte ou apresentador) x 2 categorias (associado e não-associado) x 3 opções (Profissional, Estudante de Pós-Graduação ou Estudante de Graduação e Professor de Ensino Básico)

Para cada uma dessas possibilidades haverá uma distinção da taxa a ser paga. Entretanto, note que em todos os casos o procedimento é o mesmo, apesar das diferenças.

Seguindo o **pilar de abstração e representação de informações**, devemos nos preocupar apenas com os dados relacionados ao problema.

Quanto a um congresso, existem diversas informações pertinentes, por exemplo: os dados referentes aos participantes, à programação do evento, aos parceiros e patrocinadores, informações de contatos e mídias sociais, etc. Entretanto, para o sistema que será utilizado para a venda das inscrições, é relevante saber apenas sobre duas entidades: os ingressos e os participantes.

Quanto aos participantes, não precisamos saber sobre a vida inteira deles, apenas informações-chave como: nome, instituição de ensino, e-mail, entre outros.

Já sobre os ingressos, podemos representá-los com as seguintes informações: preço, tipo de inscrição, categoria, data, etc.

Por fim, ao estruturar um algoritmo para a resolução desse problema, podemos destrinchar o passo a passo para a concretização da venda.

1. Primeiro passo: Apresentar um formulário de inscrição para o usuário
2. Segundo passo: Ler os dados inseridos no formulário
3. Terceiro passo: Identificar tipo de inscrição a ser realizada de acordo o perfil selecionado
 - a. Consultar na base de dados do valor da taxa para esse tipo de inscrição, de acordo com as combinações de opções escolhida
4. Quarto passo: Apresentar o valor da inscrição para o usuário
5. Quinto passo: Realizar a venda da inscrição
 - a. Solicitar ao usuário a forma de pagamento
 - b. Confirmar dados da compra
 - c. Finalizar pagamento

Observação: por motivos didáticos, fizemos uma simplificação dos passos do algoritmo, uma vez que o problema real teria maiores complicações.

Considerações Finais

Este material desenvolvido por Fernanda Macedo de Sousa e João Lucas Azevedo Yamin R. da Cunha enquanto graduandos do Bacharelado de Ciência da Computação sob orientação da Prof. Dra. Maristela Terto de Holanda.

Foi baseado no curso Computational Thinking for Problem Solving do Coursera oferecido pela Universidade da Pensilvânia e ministrado por Susan Davidson.

Referências

Categorias e valores - Congresso Nacional de Matemática Aplicada e Computacional. Disponível em:
<http://www.cnmac.org.br/novo/index.php/CNMAC/conteudo/2021/48/93>. Acesso em: 23/07/2021.

Computational Thinking for Problem Solving - Coursera. Disponível em:
<https://www.coursera.org/learn/computational-thinking-problem-solving>. Acesso em: 23/07/2021.