

Welton Dias de Lima

  /uniceplac
uniceplac.edu.br

PENSAMENTO COMPUTACIONAL

Abstrações

Gama, DF, 04 de Maio de 2021.



UNICEPLAC
CENTRO UNIVERSITÁRIO

CENTRO UNIVERSITÁRIO APPARECIDO DOS SANTOS - UNICEPLAC

Dados Internacionais de Catalogação na Publicação (CIP)

L732p

Lima, Welton Dias de.

Pensamento computacional: abstrações. Gama, DF:
UNICEPLAC, 2021.

20 p.

1. Computação. 2. Pensamento computacional. 3.
Abstrações. I. Título.

CDU: 004

Pensamento computacional: abstrações

Marcelo da Silva dos Santos

OBJETIVOS DE APRENDIZAGEM

- > Conceituar abstração.
- > Identificar abstrações em camada.
- > Classificar os riscos e as limitações do uso de abstrações.

PENSAMENTO COMPUTACIONAL

TÓPICOS DA AULA:

- a) Abstração e pensamento computacional.
- b) Abstrações em camadas.
- c) Riscos e limitações do uso de abstrações.

Introdução

A abstração é conceito essencial do pensamento computacional, capaz de criar representações simplificadas de qualquer ser vivo ou objeto. Quando em excesso, os detalhes sobre a entidade podem desviar o foco da solução ou trazer complexidade extra para o processamento. Modelos abstratos são muito utilizados em várias áreas do conhecimento. Por exemplo, um cientista cria modelos simplificados ignorando condições climáticas, atrito e resistência do ar, e mesmo que não reflitam completamente a realidade, são capazes de simular situações específicas e resolver grandes questões.

Abstrair é uma habilidade do ser humano, que consiste na capacidade de expressar algo de maneira concisa, eliminando detalhes desnecessários. Saber trabalhar com diferentes camadas de abstração é um requisito para o profissional da computação e, ao mesmo tempo, um constante desafio. Diversos riscos e limitações a seu uso estão presentes, algumas vezes não sendo possível solucioná-los, apenas adaptar a abstração para absorvê-los.

PENSAMENTO COMPUTACIONAL

a) Abstração e pensamento computacional.

O pensamento computacional pode ser compreendido com uma estratégia para resolução de problemas utilizando conceitos computacionais (WING, 2006).

A proposta é utilizar a forma de raciocinar de um cientista da computação, mas não atrelar a solução ao uso do computador.

Até porque a maioria desses conceitos foram originados antes mesmo da criação do computador eletrônico, e sua aplicação transcende a ciência da computação (RILEY; HUNT, 2014).

Além de estimular o pensamento lógico e sistemático, também promove o pensamento crítico, estratégico e criativo.

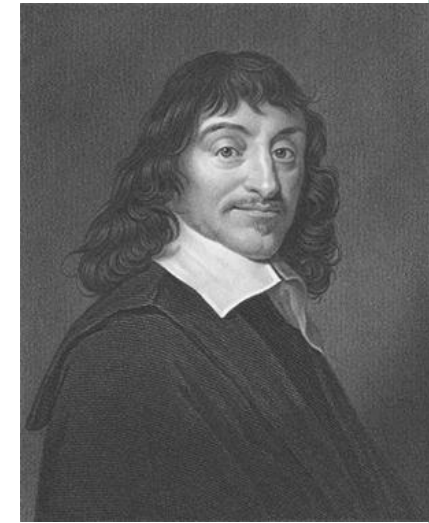
Como comentado anteriormente, **o pensamento computacional faz uso de conceitos emprestados da computação**, que dão suporte para a definição de seus fundamentos, também chamados de pilares do pensamento computacional: **a decomposição, o reconhecimento de padrões, a abstração e os algoritmos** (WING, 2006).

PENSAMENTO COMPUTACIONAL

a) Abstração e pensamento computacional.



O método científico proposto por Descartes e que predominou até o final do século XIX e o início do século XX, ficou conhecido como “Determinismo Mecanicista” e se resume aos seguintes princípios: **o conhecimento é o resultado da captura de verdades por um sujeito sobre um objeto; o sujeito percebe o objeto a partir de exercícios sensitivos e racionais que devem ser organizados de forma metodológica a fim de se obter o conhecimento verdadeiro; o objeto é separado do observador; conhecer o objeto é igual a dominá-lo; para conhecer o todo, basta conhecer as partes**; o método cartesiano, nesse sentido, implica em uma simplificação onde o objetivo é encontrar lei universal que explique todas as coisas; o mundo pode ser expresso por meio de equações matemáticas; o mundo deve ser compreendido, dominado e modificado em favor do homem.



PENSAMENTO COMPUTACIONAL

a) Abstração e pensamento computacional.

Conceito de abstração Segundo Wing (2006). A abstração é um conceito-chave tanto da computação quanto do pensamento computacional, chegando a descrever a ciência da computação como “a automação da abstração”.

Quando precisamos que o computador execute uma tarefa, devemos explicar somente as informações necessárias para que ele consiga executar, até porque, se não filtrarmos (e dependendo do nível de detalhes), podemos confundi-lo com o excesso de informações e desviar do objetivo principal.

Esse processo de selecionar quais informações são mais ou menos relevantes é o que chamamos de abstração. Segundo Guttag (2013), a essência da abstração é manter as informações relevantes de determinado contexto e omitir todo o resto. Lembrando que esse “relevante” sempre vai depender da finalidade da tarefa.

PENSAMENTO COMPUTACIONAL

a) Abstração e pensamento computacional.

O pensamento abstrato é uma habilidade cognitiva importante do ser humano e que aplicamos a todo momento, sem perceber. Criamos abstrações, ou simplificações, para tudo que está a nosso redor, desde conceitos simples até artefatos complexos. Por exemplo, pense na palavra “número”: só de ouvir esse termo você já sabe do que se trata, mesmo sem ouvir outros detalhes. Os números são abstrações para representar a quantificação de algo.



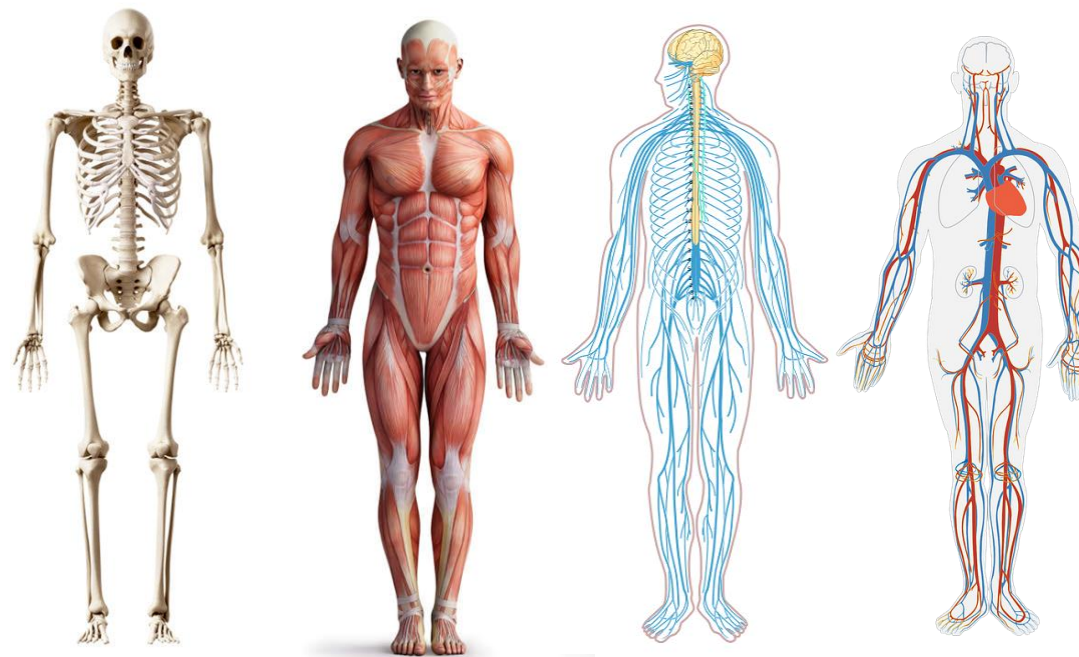
Figura 2. Outras representações do mesmo mapa metroviário com diferentes níveis de abstração.

Fonte: Adaptada de Beecher (2017).

PENSAMENTO COMPUTACIONAL

B) Abstrações em camadas

Como visto anteriormente, a abstração é a capacidade de identificar os detalhes importantes do que se está observando, mas a quantidade desses “detalhes” que atribuímos ao elemento depende do objetivo em que será empregado. O volume de informações vai impactar significativamente a representação. Também vimos que é possível criar diferentes versões de representação para o mesmo elemento, com níveis diferentes de detalhamento.



PENSAMENTO COMPUTACIONAL

B) Abstrações em camadas

Note, na Figura , uma sequência de estudos do pintor Pablo Picasso, com diversas representações do mesmo elemento (o touro), somente diferindo o nível de detalhes.



PENSAMENTO COMPUTACIONAL

B) Abstrações em camadas

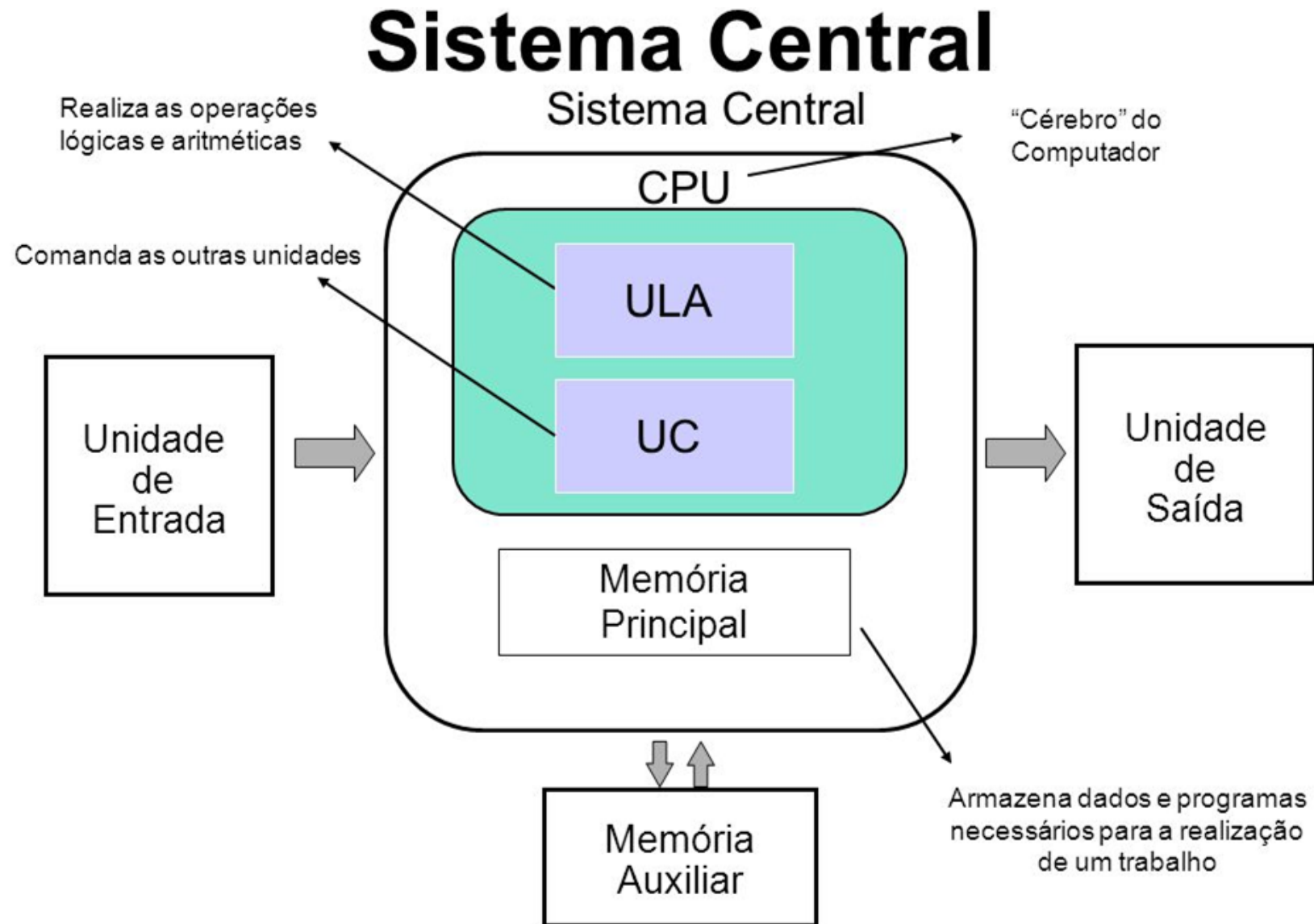
Essa diferença entre os níveis de abstração é conhecida como camadas de abstração.

Cada camada representa uma quantidade diferente de informações e, cada vez que a trocamos, podemos aumentar os detalhes (reduzindo a abstração) ou adicionar mais camadas de abstração para suprimir detalhes (aumentando a abstração).

Um exemplo mais próximo ao contexto computacional, quando você pressiona uma tecla qualquer do teclado do computador. Você precisa saber todos os processos (sinais, interrupções, pulsos elétricos) que estão ocorrendo naquele instante ou só necessita que o caractere seja apresentado no monitor? Caso você precise manipular o comportamento do software para reagir ao caractere pressionado, vai remover algumas camadas de abstração (adicionar detalhes) até que chegue a um modelo que seja adequado ao contexto.

PENSAMENTO COMPUTACIONAL

B) Abstrações em camadas



PENSAMENTO COMPUTACIONAL

B) Abstrações em camadas

Computacionalmente, diversas áreas empregam camadas de abstrações em seus processos, como as elencadas a seguir.

- **Modelagem de banco de dados:** modelo entidade relacionamento utiliza três camadas de abstração diferentes, um nível para cada estágio de maturidade da implementação.
- **Análise de sistemas:** em processos de análise de sistemas, são utilizados diagramas e documentos com diversos níveis de abstração, como o diagrama de caso de uso (alta abstração) da UML e o documento de descrição do caso de uso (baixa abstração).
- **Programação:** em orientação a objetos, por exemplo, entidades são representadas como classes abstratas ou interfaces (maior abstração que a própria classe abstrata), entre outras. Aqui, ainda se pode destacar o POJO (Plain Old Java Object), outro tipo de classe que utiliza o conceito de abstração para modelagem (o mais simples possível) de entidades.
- **Webdesign:** assim como o analista de sistemas, utiliza inúmeros artefatos com níveis de abstração diferente (*wireframes*, protótipos, entre outros) em estágios de maturidade diversos.
- **Hardware e redes:** na construção de *hardware* e de outros equipamentos de infraestrutura de redes, utilizam camadas e abstrações em suas representações.

PENSAMENTO COMPUTACIONAL

c) Riscos e limitações do uso de abstrações

Como visto nas seções anteriores, a abstração, além de função cognitiva inata do ser humano, é um dos pilares do pensamento computacional, permitindo que possamos representar elementos do mundo real de forma simplificada e, assim, utilizá-los na solução de problemas.

Também foi apresentado como uma abstração da realidade pode ser representada de múltiplas formas, dependendo da quantidade de camadas utilizadas em sua construção. O fato é que abstrações são conceitos muito ligados à experiência e à perspectiva do observador.

Como comentado anteriormente, **cada analista descreve uma abstração levando em conta as particularidades importantes para ele**. Outro exercício de abstração: como você descreveria um gato?

O exercício serve para recordar que não existe uma definição mais correta, mas a mais adequada para determinado cenário.

PENSAMENTO COMPUTACIONAL

c) Riscos e limitações do uso de abstrações

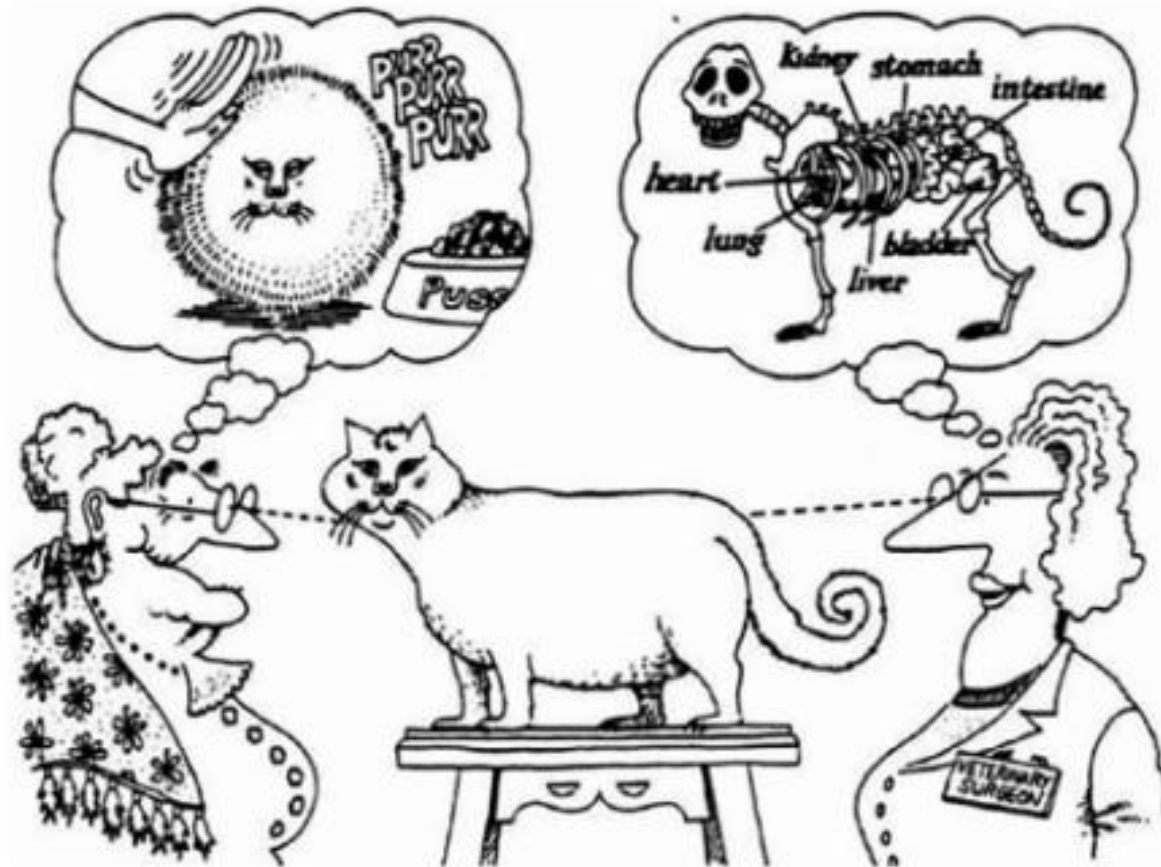


Figura 4. Experiência do analista e diferentes perspectivas podem influenciar o resultado da abstração.

Fonte: Adaptada de Booch (2004).

PENSAMENTO COMPUTACIONAL

c) Riscos e limitações do uso de abstrações

As abstrações são essenciais para o pensamento computacional, mas é necessário atenção para que elas possam ser realmente úteis. A definição de uma abstração é uma tarefa complexa, de forma que o risco de erros está sempre presente. Um dos problemas mais comuns é o vazamento de abstração, que ocorre quando uma entidade vaza detalhes de sua construção, como propriedades ou comportamento específico de implementação (VAN DEURSEN; SEEMANN, 2019).

O vazamento da abstração é algo que não está mas deveria estar oculto, que quem vai consumir não deveria saber ou ter que lidar e acaba tendo que fazer isso. Então há vazamento quando o consumidor tem que lidar com coisas que poderiam ter sido escondidas dele sem prejuízos.

PENSAMENTO COMPUTACIONAL

c) Riscos e limitações do uso de abstrações

Agora é com você!!!

Pensamento lógico e capacidade de abstração são habilidades úteis na elaboração de programas de computador. Tais habilidades podem ser desenvolvidas e melhoradas através da resolução de exercícios que necessitam apenas de raciocínio e bom senso.

Considere a realização de uma pesquisa com 1 000 pessoas para obtenção das seguintes informações: o valor da maior altura; o valor da menor altura; a média das alturas; quantas pessoas têm altura inferior à média das alturas.

Considere, ainda, que um programador foi selecionado para desenvolver um modelo de código que soluciona o problema automatizando a coleta das alturas e a geração das informações.

Com base nas informações apresentadas, desenvolva o código adequado para resolver o problema usando pseudocódigo ou uma linguagem de programação.

REFERÊNCIAS

- D'AMBROSIO, Ubiratan. Educação Matemática da teoria a pratica. Campinas, São Paulo: Papirus, 1996.
- DANTE, L.R. Didática da Resolução de Problemas de Matemática. 2ªed. São Paulo: Ática, 1998.
- DANTE, L. R. Didática da resolução de problemas. São Paulo: Ática, 2005
- NUNES, C.B & SOUZA, A.C.P. A Resolução de problemas como metodologia de ensino aprendizagem-avaliação de Matemática em sala de aula. UNESP, Rio Claro - SP. Disponível em: www.sbem.com.br/files/ix_enem/Minicurso/Resumos/MC65873300534R.doc. Acesso em: 31 de maio de 2015.
- ONUCHIC, L. R. Ensino-Aprendizagem de Matemática através da Resolução de Problemas. In: BICUDO, M. A. V (org.). Pesquisa em Educação Matemática: Concepções e Perspectivas. São Paulo: Ed. Unesp, p.199-220, 1999.

Obrigado (a)!

