

Hellon Canella Machado

**O Uso do Pensamento Computacional como
Recurso para o Desenvolvimento da
Aprendizagem Científica.**

Rio de Janeiro. Brasil

2018

1 Resultados de comandos

Isto é uma sinopse de capítulo. A ABNT não traz nenhuma normatização a respeito desse tipo de resumo, que é mais comum em romances e livros técnicos.

1.1 Codificação dos arquivos: UTF8

A codificação de todos os arquivos do `abnTeX2` é UTF8. É necessário que você utilize a mesma codificação nos documentos que escrever, inclusive nos arquivos de base bibliográficas `|.bib|`.

1.2 Citações diretas

Utilize o ambiente `citacao` para incluir citações diretas com mais de três linhas:

As citações diretas, no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem aspas. No caso de documentos datilografados, deve-se observar apenas o recuo (??, 5.3).

Use o ambiente assim:

```
\begin{citacao}
```

As citações diretas, no texto, com mais de três linhas [...] deve-se observar apenas o recuo `\cite[5.3]{NBR10520:2002}`.

```
\end{citacao}
```

O ambiente `citacao` pode receber como parâmetro opcional um nome de idioma previamente carregado nas opções da classe (??). Nesse caso, o texto da citação é automaticamente escrito em itálico e a hifenização é ajustada para o idioma selecionado na opção do ambiente. Por exemplo:

```
\begin{citacao}[english]
```

Text in English language in italic with correct hyphenation.

```
\end{citacao}
```

Tem como resultado:

Text in English language in italic with correct hyphenation.

Citações simples, com até três linhas, devem ser incluídas com aspas. Observe que em `LATEX` as aspas iniciais são diferentes das finais: “Amor é fogo que arde sem se ver”.

1.3 Notas de rodapé

As hello de rodapé são detalhadas pela NBR 14724:2011 na seção 5.2.1^{1,2,3}.

1.4 Tabelas

?? (?? apud ??, ??, p. 2–3) ?? (??, p. 32–34) A ?? é um exemplo de tabela construída em L^AT_EX.

Tabela 1 – Níveis de investigação.

Nível de Inves- tigação	Insumos	Sistemas de Investigação	Produtos
Meta-nível	Filosofia da Ciência	Epistemologia	Paradigma
Nível do objeto	Paradigmas do metanível e evidências do nível inferior	Ciência	Teorias e modelos
Nível inferior	Modelos e métodos do nível do objeto e problemas do nível inferior	Prática	Solução de problemas

Fonte: ??)

Já a ?? apresenta uma tabela criada conforme o padrão do ??) requerido pelas normas da ABNT para documentos técnicos e acadêmicos.

Tabela 2 – Um Exemplo de tabela alinhada que pode ser longa ou curta, conforme padrão IBGE.

Nome	Nascimento	Documento
Maria da Silva	11/11/1111	111.111.111-11
João Souza	11/11/2111	211.111.111-11
Laura Vicuña	05/04/1891	3111.111.111-11

Fonte – Produzido pelos autores.

Nota – Esta é uma nota, que diz que os dados são baseados na regressão linear.

Anotações – Uma anotação adicional, que pode ser seguida de várias outras.

¹ As notas devem ser digitadas ou datilografadas dentro das margens, ficando separadas do texto por um espaço simples de entre as linhas e por filete de 5 cm, a partir da margem esquerda. Devem ser alinhadas, a partir da segunda linha da mesma nota, abaixo da primeira letra da primeira palavra, de forma a destacar o expoente, sem espaço entre elas e com fonte menor ??, 5.2.1).

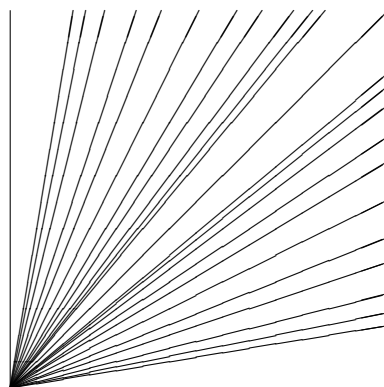
² Caso uma série de notas sejam criadas sequencialmente, o abnT_EX2 instrui o L^AT_EX para que uma vírgula seja colocada após cada número do expoente que indica a nota de rodapé no corpo do texto.

³ Verifique se os números do expoente possuem uma vírgula para dividi-los no corpo do texto.

1.5 Figuras

Figuras podem ser criadas diretamente em \LaTeX , como o exemplo da ??.

Figura 1 – A delimitação do espaço



Fonte: os autores

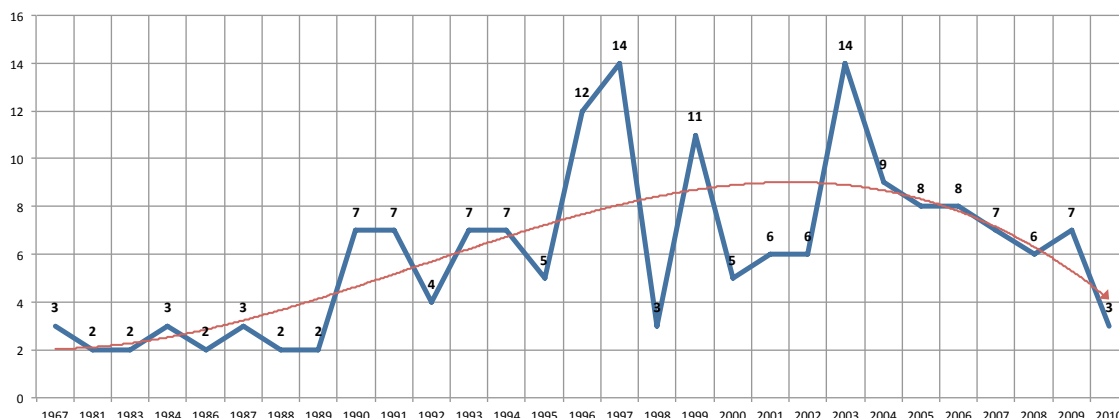
Ou então figuras podem ser incorporadas de arquivos externos, como é o caso da ??. Se a figura que se incluir se tratar de um diagrama, um gráfico ou uma ilustração que você mesmo produza, priorize o uso de imagens vetoriais no formato PDF. Com isso, o tamanho do arquivo final do trabalho será menor, e as imagens terão uma apresentação melhor, principalmente quando impressas, uma vez que imagens vetoriais são perfeitamente escaláveis para qualquer dimensão. Nesse caso, se for utilizar o Microsoft Excel para produzir gráficos, ou o Microsoft Word para produzir ilustrações, exporte-os como PDF e os incorpore ao documento conforme o exemplo abaixo. No entanto, para manter a coerência no uso de software livre (já que você está usando \LaTeX e abnTeX2), teste a ferramenta **InkScape** (<<http://inkscape.org/>>). Ela é uma excelente opção de código-livre para produzir ilustrações vetoriais, similar ao CorelDraw ou ao Adobe Illustrator. De todo modo, caso não seja possível utilizar arquivos de imagens como PDF, utilize qualquer outro formato, como JPEG, GIF, BMP, etc. Nesse caso, você pode tentar aprimorar as imagens incorporadas com o software livre **Gimp** (<<http://www.gimp.org/>>). Ele é uma alternativa livre ao Adobe Photoshop.

1.5.1 Figuras em *minipages*

Minipages são usadas para inserir textos ou outros elementos em quadros com tamanhos e posições controladas. Veja o exemplo da ?? e da ??.

Observe que, segundo a ??, seções 4.2.1.10 e 5.8), as ilustrações devem sempre ter numeração contínua e única em todo o documento:

Figura 2 – Gráfico produzido em Excel e salvo como PDF



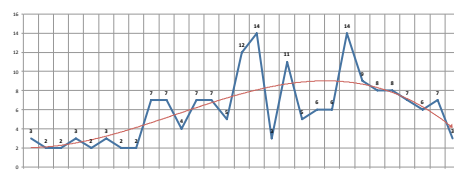
Fonte: ??, p. 24)

Figura 3 – Imagem 1 da minipage



Fonte: Produzido pelos autores

Figura 4 – Grafico 2 da minipage



Fonte: ??, p. 24)

Qualquer que seja o tipo de ilustração, sua identificação aparece na parte superior, precedida da palavra designativa (desenho, esquema, fluxograma, fotografia, gráfico, mapa, organograma, planta, quadro, retrato, figura, imagem, entre outros), seguida de seu número de ordem de ocorrência no texto, em algarismos arábicos, travessão e do respectivo título. Após a ilustração, na parte inferior, indicar a fonte consultada (elemento obrigatório, mesmo que seja produção do próprio autor), legenda, notas e outras informações necessárias à sua compreensão (se houver). A ilustração deve ser citada no texto e inserida o mais próximo possível do trecho a que se refere. (??, seções 5.8)

1.6 Expressões matemáticas

Use o ambiente `equation` para escrever expressões matemáticas numeradas:

$$\forall x \in X, \quad \exists y \leq \epsilon \quad (1.1)$$

Escreva expressões matemáticas entre `$` e `$`, como em $\lim_{x \rightarrow \infty} \exp(-x) = 0$, para que fiquem na mesma linha.

Também é possível usar colchetes para indicar o início de uma expressão matemática

que não é numerada.

$$\left| \sum_{i=1}^n a_i b_i \right| \leq \left(\sum_{i=1}^n a_i^2 \right)^{1/2} \left(\sum_{i=1}^n b_i^2 \right)^{1/2}$$

Consulte mais informações sobre expressões matemáticas em <https://code.google.com/p/abntex2/wiki/Referencias>.

1.7 Enumerações: alíneas e subalíneas

Quando for necessário enumerar os diversos assuntos de uma seção que não possua título, esta deve ser subdividida em alíneas (??, 4.2):

- a) os diversos assuntos que não possuam título próprio, dentro de uma mesma seção, devem ser subdivididos em alíneas;
- b) o texto que antecede as alíneas termina em dois pontos;
- c) as alíneas devem ser indicadas alfabeticamente, em letra minúscula, seguida de parêntese. Utilizam-se letras dobradas, quando esgotadas as letras do alfabeto;
- d) as letras indicativas das alíneas devem apresentar recuo em relação à margem esquerda;
- e) o texto da alínea deve começar por letra minúscula e terminar em ponto-e-vírgula, exceto a última alínea que termina em ponto final;
- f) o texto da alínea deve terminar em dois pontos, se houver subalínea;
- g) a segunda e as seguintes linhas do texto da alínea começa sob a primeira letra do texto da própria alínea;
- h) subalíneas (??, 4.3) devem ser conforme as alíneas a seguir:
 - as subalíneas devem começar por travessão seguido de espaço;
 - as subalíneas devem apresentar recuo em relação à alínea;
 - o texto da subalínea deve começar por letra minúscula e terminar em ponto-e-vírgula. A última subalínea deve terminar em ponto final, se não houver alínea subsequente;
 - a segunda e as seguintes linhas do texto da subalínea começam sob a primeira letra do texto da própria subalínea.
- i) no abnT_EX2 estão disponíveis os ambientes `incisos` e `subalíneas`, que em suma são o mesmo que se criar outro nível de `alíneas`, como nos exemplos à seguir:
 - *Um novo inciso em itálico;*

- j) Alínea em **negrito**:
 - *Uma subalínea em itálico*;
 - *Uma subalínea em itálico e sublinhado*;
- k) Última alínea com *ênfase*.

1.8 Espaçamento entre parágrafos e linhas

O tamanho do parágrafo, espaço entre a margem e o início da frase do parágrafo, é definido por:

```
\setlength{\parindent}{1.3cm}
```

Por padrão, não há espaçamento no primeiro parágrafo de cada início de divisão do documento (??). Porém, você pode definir que o primeiro parágrafo também seja indentado, como é o caso deste documento. Para isso, apenas inclua o pacote `indentfirst` no preâmbulo do documento:

```
\usepackage{indentfirst}      % Indenta o primeiro parágrafo de cada seção.
```

O espaçamento entre um parágrafo e outro pode ser controlado por meio do comando:

```
\setlength{\parskip}{0.2cm}  % tente também \onelineskip
```

O controle do espaçamento entre linhas é definido por:

```
\OnehalfSpacing      % espaçamento um e meio (padrão);
\DoubleSpacing        % espaçamento duplo
\SingleSpacing        % espaçamento simples
```

Para isso, também estão disponíveis os ambientes:

```
\begin{SingleSpace} ... \end{SingleSpace}
\begin{Spacing}{hfactori} ... \end{Spacing}
\begin{OnehalfSpace} ... \end{OnehalfSpace}
\begin{OnehalfSpace*} ... \end{OnehalfSpace*}
\begin{DoubleSpace} ... \end{DoubleSpace}
\begin{DoubleSpace*} ... \end{DoubleSpace*}
```

Para mais informações, consulte ??, p. 47-52 e 135).

1.9 Inclusão de outros arquivos

É uma boa prática dividir o seu documento em diversos arquivos, e não apenas escrever tudo em um único. Esse recurso foi utilizado neste documento. Para incluir diferentes arquivos em um arquivo principal, de modo que cada arquivo incluído fique em uma página diferente, utilize o comando:

```
\include{documento-a-ser-incluido} % sem a extensão .tex
```

Para incluir documentos sem quebra de páginas, utilize:

```
\input{documento-a-ser-incluido} % sem a extensão .tex
```

1.10 Compilar o documento L^AT_EX

Geralmente os editores L^AT_EX, como o TeXlipse⁴, o Texmaker⁵, entre outros, compilam os documentos automaticamente, de modo que você não precisa se preocupar com isso.

No entanto, você pode compilar os documentos L^AT_EX usando os seguintes comandos, que devem ser digitados no *Prompt de Comandos* do Windows ou no *Terminal* do Mac ou do Linux:

```
pdflatex ARQUIVO_PRINCIPAL.tex
bibtex ARQUIVO_PRINCIPAL.aux
makeindex ARQUIVO_PRINCIPAL.idx
makeindex ARQUIVO_PRINCIPAL.nlo -s nomencl.ist -o ARQUIVO_PRINCIPAL.nls
pdflatex ARQUIVO_PRINCIPAL.tex
pdflatex ARQUIVO_PRINCIPAL.tex
```

1.11 Remissões internas

Ao nomear a ?? e a ??, apresentamos um exemplo de remissão interna, que também pode ser feita quando indicamos o ??, que tem o nome ???. O número do capítulo indicado é ??, que se inicia à página ??⁶. Veja a ?? para outros exemplos de remissões internas entre seções, subseções e subsubseções.

O código usado para produzir o texto desta seção é:

⁴ <<http://texlipse.sourceforge.net/>>

⁵ <<http://www.xmlmath.net/texmaker/>>

⁶ O número da página de uma remissão pode ser obtida também assim: ??.

Ao nomear a `\autoref{tab-nivinv}` e a `\autoref{fig_circulo}`, apresentamos um exemplo de remissão interna, que também pode ser feita quando indicamos o `\autoref{cap_exemplos}`, que tem o nome `\emph{\nameref{cap_exemplos}}`. O número do capítulo indicado é `\ref{cap_exemplos}`, que se inicia à `\autopageref{cap_exemplos}`. O número da página de uma remissão pode ser obtida também assim:

`\pageref{cap_exemplos}`}.}

Veja a `\autoref{sec-divisoões}` para outros exemplos de remissões internas entre seções, subseções e subsubseções.

1.12 Divisões do documento: seção

Esta seção testa o uso de divisões de documentos. Esta é a ???. Veja a ???.

1.12.1 Divisões do documento: subseção

Isto é uma subseção. Veja a ???, que é uma `subsubsection` do `LATEX`, mas é impressa chamada de “subseção” porque no Português não temos a palavra “subsubseção”.

1.12.1.1 Divisões do documento: subsubseção

Isto é uma subsubseção.

1.12.1.2 Divisões do documento: subsubseção

Isto é outra subsubseção.

1.12.2 Divisões do documento: subseção

Isto é uma subseção.

1.12.2.1 Divisões do documento: subsubseção

Isto é mais uma subsubseção da ???.

1.12.2.1.1 Esta é uma subseção de quinto nível

Esta é uma seção de quinto nível. Ela é produzida com o seguinte comando:

```
\subsubsubsection{Esta é uma subseção de quinto
nível}\label{sec-exemplo-subsubsubsection}
```

1.12.2.1.2 Esta é outra subseção de quinto nível

Esta é outra seção de quinto nível.

1.12.2.1.3 Este é um parágrafo numerado

Este é um exemplo de parágrafo nomeado. Ele é produzida com o comando de parágrafo:

```
\paragraph{Este é um parágrafo nomeado}\label{sec-exemplo-paragrafo}
```

A numeração entre parágrafos numerados e subsubsubseções são contínuas.

1.12.2.1.4 Esta é outro parágrafo numerado

Esta é outro parágrafo nomeado.

1.13 Este é um exemplo de nome de seção longo. Ele deve estar alinhado à esquerda e a segunda e demais linhas devem iniciar logo abaixo da primeira palavra da primeira linha

Isso atende à norma ??, seções de 5.2.2 a 5.2.4) e ??, seções de 3.1 a 3.8).

1.14 Diferentes idiomas e hifenizações

Para usar hifenizações de diferentes idiomas, inclua nas opções do documento o nome dos idiomas que o seu texto contém. Por exemplo (para melhor visualização, as opções foram quebradas em diferentes linhas):

```
\documentclass[  
12pt,  
openright,  
twoside,  
a4paper,  
english,  
french,  
spanish,  
brazil  
{abntex2}
```

O idioma português-brasileiro (`brazil`) é incluído automaticamente pela classe `abntex2`. Porém, mesmo assim a opção `brazil` deve ser informada como a última opção da classe para que todos os pacotes reconheçam o idioma. Vale ressaltar que a última opção de idioma é a utilizada por padrão no documento. Desse modo, caso deseje escrever um texto em inglês que tenha citações em português e em francês, você deveria usar o preâmbulo como abaixo:

```
\documentclass[
12pt,
openright,
twoside,
a4paper,
french,
brazil,
english
]{abntex2}
```

A lista completa de idiomas suportados, bem como outras opções de hifenização, estão disponíveis em ??, p. 5-6).

Exemplo de hifenização em inglês⁷:

Text in English language. This environment switches all language-related definitions, like the language specific names for figures, tables etc. to the other language. The starred version of this environment typesets the main text according to the rules of the other language, but keeps the language specific string for ancillary things like figures, in the main language of the document. The environment hyphenrules switches only the hyphenation patterns used; it can also be used to disallow hyphenation by using the language name ‘nohyphenation’.

Exemplo de hifenização em francês⁸:

Texte en français. Pas question que Twitter ne vienne faire une concurrence déloyale à la traditionnelle fumée blanche qui marque l’élection d’un nouveau pape. Pour éviter toute fuite précoce, le Vatican a donc pris un peu d’avance, et a déjà interdit aux cardinaux qui prendront part au vote d’utiliser le réseau social, selon Catholic News Service. Une mesure valable surtout pour les neuf cardinaux – sur les 117 du conclave – pratiquants très actifs de Twitter, qui auront interdiction pendant toute la période de se connecter à leur compte.

⁷ Extraído de: <<http://en.wikibooks.org/wiki/LaTeX/Internationalization>>

⁸ Extraído de: <<http://bigbrowser.blog.lemonde.fr/2013/02/17/tu-ne-tweeteras-point-le-vatican-interdit-aux-cardinaux-d>>

Pequeno texto em espanhol⁹:

Decenas de miles de personas ovacionan al pontífice en su penúltimo ángelus dominical, el primero desde que anunciase su renuncia. El Papa se centra en la crítica al materialismo.

O idioma geral do texto por ser alterado como no exemplo seguinte:

```
\selectlanguage{english}
```

Isso altera automaticamente a hifenização e todos os nomes constantes de referências do documento para o idioma inglês. Consulte o manual da classe (??) para obter orientações adicionais sobre internacionalização de documentos produzidos com abnT_EX2.

A ?? descreve o ambiente `citacao` que pode receber como parâmetro um idioma a ser usado na citação.

1.15 Consulte o manual da classe abntex2

Consulte o manual da classe `abntex2` (??) para uma referência completa das macros e ambientes disponíveis.

Além disso, o manual possui informações adicionais sobre as normas ABNT observadas pelo abnT_EX2 e considerações sobre eventuais requisitos específicos não atendidos, como o caso da ??, seção 5.2.2), que especifica o espaçamento entre os capítulos e o início do texto, regra propositalmente não atendida pelo presente modelo.

1.16 Referências bibliográficas

A formatação das referências bibliográficas conforme as regras da ABNT são um dos principais objetivos do abnT_EX2. Consulte os manuais ??) e ??) para obter informações sobre como utilizar as referências bibliográficas.

1.16.1 Acentuação de referências bibliográficas

Normalmente não há problemas em usar caracteres acentuados em arquivos bibliográficos (*.bib). Porém, como as regras da ABNT fazem uso quase abusivo da conversão para letras maiúsculas, é preciso observar o modo como se escreve os nomes dos autores. Na ?? você encontra alguns exemplos das conversões mais importantes. Preste atenção especial para ‘ç’ e ‘í’ que devem estar envoltos em chaves. A regra geral é sempre usar a acentuação neste modo quando houver conversão para letras maiúsculas.

⁹ Extraído de: <http://internacional.elpais.com/internacional/2013/02/17/actualidad/1361102009_913423.html>

Tabela 3 – Tabela de conversão de acentuação.

acento	bibtex
à á ã	\‘a \’a \~a
í	{\’\i}
ç	{\c c}

1.17 Precisa de ajuda?

Consulte a FAQ com perguntas frequentes e comuns no portal do abnT_EX2: <<https://code.google.com/p/abntex2/wiki/FAQ>>.

Inscreva-se no grupo de usuários L^AT_EX: <<http://groups.google.com/group/latex-br>>, tire suas dúvidas e ajude outros usuários.

Participe também do grupo de desenvolvedores do abnT_EX2: <<http://groups.google.com/group/abntex2>> e faça sua contribuição à ferramenta.

1.18 Você pode ajudar?

Sua contribuição é muito importante! Você pode ajudar na divulgação, no desenvolvimento e de várias outras formas. Veja como contribuir com o abnT_EX2 em <<https://code.google.com/p/abntex2/wiki/ComoContribuir>>.

1.19 Quer customizar os modelos do abnT_EX2 para sua instituição ou universidade?

Veja como customizar o abnT_EX2 em: <<https://code.google.com/p/abntex2/wiki/ComoCustomizar>>.

2 Computadores e a prática científica

A evolução da computação nas últimas décadas, aliada ao seu barateamento, tem produzido impactos notáveis tanto na academia quanto na indústria. A disponibilidade de dispositivos mais baratos e com maior capacidade de processamento e armazenamento tem tornado a computação ubíqua.

Na academia, esse fenômeno tem favorecido o surgimento de novas estratégias para exploração de fenômenos. Até meados do século 20, todo progresso científico foi conduzido apenas por interações entre atividades experimentais e analíticas¹. O surgimento da computação e o seu desenvolvimento desde então trouxe consigo novas formas de fazer ciência, tais como a simulação e a modelagem computacional, e mais recentemente, a mineração de dados e o aprendizado de máquina, úteis para a análise de grande volume de informação (????).

O uso de simulações numéricas, por exemplo, se justifica ao permitir que um grande número de fenômenos muito complexos sejam analiticamente tratáveis. Em muitos casos essa é única forma de exploração possível. Mesmo na mecânica newtoniana mais simples, é possível resolver exatamente, apenas, o problema de dois corpos. Para $N \geq 3$, soluções numéricas são necessárias. Da astronomia podemos retirar alguns exemplos, como a formação de estrelas e galáxias e explosão estelares - de modo geral, qualquer evento envolvendo turbulência (??).

O uso de métodos computacionais, tal como a simulação, tem expandido a abrangência de sistemas não lineares que tem sido explorados pelos modelos matemáticos e computacionais. Como lembra ??), campos da ciência estão experimentando um renascimento de abordagens experimentais em razão do acesso facilitado a mais poder computacional.

O autor destaca que num passado recente, para muitos pesquisadores apenas o estudo de sistemas determinísticos era viável, tendo o termo ‘não-linear’, praticamente, o sinônimo de ‘insolúvel’. Havia desse modo a propensão à investigação computacional apenas de sistemas lineares. Esse quadro era especialmente verdade para muitas pesquisas em biologia e química.

Esse processo ganha especial relevância ao nos darmos conta da natureza caótica da ampla maioria dos fenômenos físicos. Sistemas lineares e determinísticos são portanto exceções, e não a regra (??).

¹ Em sentido mais amplo, quando mencionamos ‘atividade analítica’ estamos nos referindo à utilização de aparato matemático para resolução teórica de problemas. Ao mesmo tempo, o termo análise também pode fazer referência ao emprego da análise matemática, ramo da matemática que lida com conceitos do cálculo diferencial, tais como diferenciação, integração, e séries infinitas.

Outros agentes de transformações da prática científica, embora mais recentes e em fase nascente, são as novas possibilidades trazidas pela abundância e pelo barateamento do armazenamento grandes volumes de dados. Vivemos uma era onde a disponibilidade de dados gerados por câmeras, sensores, execução de simulações e registro de interação humana crescem exponencialmente.

Nesse cenário, como ressalta ??), o foco de valores tem mudado da propriedade de dados ou de instrumentos para reuni-los para a propriedade de conhecimento e ideias que tornam possíveis a extração de significados desse volume de informação.

A abundância traz consigo muitos desafios. A taxa com que cientistas e engenheiros têm coletado e produzido dados vem exigindo avanços nas estratégias de análise. O acúmulo chegou a um nível de complexidade que, certo modo, tem sido impossível fazer qualquer tipo de investigação superficial utilizando técnicas convencionais, baseadas na percepção humana. Lidar com esse conjunto desestruturado e rico de dados, extraíndo dele significado, tem sido uma das batalhas da atual revolução científica e industrial (??). Nesse contexto o emprego de técnicas de aprendizado de máquina é essencial.

Em linhas gerais, o aprendizado de máquina (do inglês: *machine learning*) baseia-se no uso algoritmos que instruem computadores como avaliar dados e deles extrair padrões e correlações, permitindo-os, de forma extraordinária, a fazer previsões. E esse processo tem um componente recursivo: quanto mais análises são feitas, mais experiência e competência são adquiridas. Ou seja, mais ‘inteligentes’ essas máquinas se tornam.

??) propõe uma visão simplificada das interações humanas que facilita o entendimento desses algoritmos. Como descreve, ao conhecemos alguém pela primeira vez, baseando-nos em modelos pessoais, somos capazes de dizer nos primeiros minutos se essa pessoa nos transmite boa ou má impressão. Para cada nova pessoa que encontramos, avaliamos algumas de suas características e as registramos. Esse processo nos permite refinar e recompor modelos sociais que irão influenciar outras percepções em interações futuras.

É exatamente nesse princípio recursivo que se fundamenta o aprendizado de máquina: categorizar dados de acordo com suas características com o fim de compor e refinar modelos.

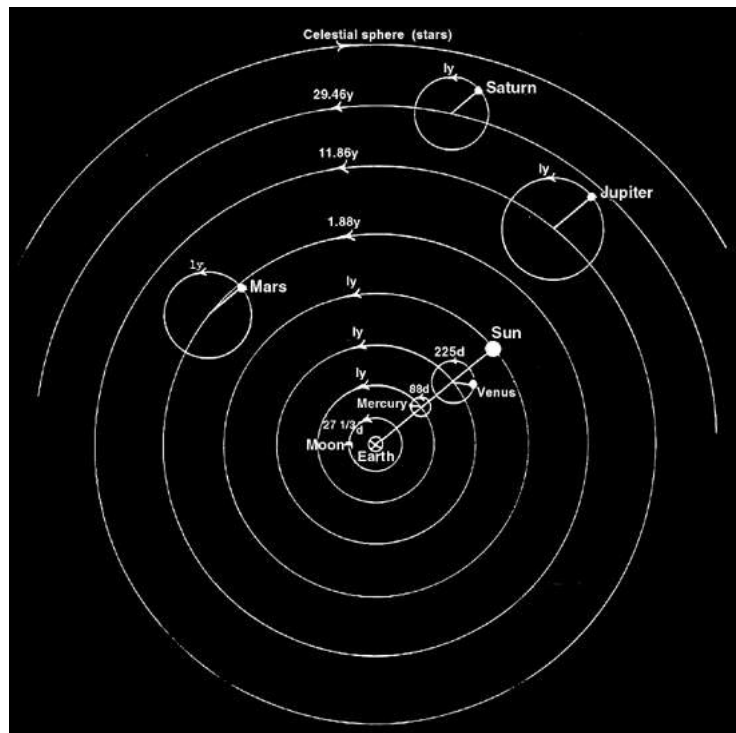
Esse processo tem se provado extremamente eficiente na predição da configuração de sistemas estocásticos. Tomemos por exemplo a necessidade de prever o tempo, onde há dominância de comportamentos turbulentos. Em um estudo recente (??, ?? apud ??, ??), mostrou-se a eficiência do uso de algoritmos de inteligência artificial para previsões ao longo de um período muito maior do que se imaginou possível. Um ponto digno de nota: o algoritmo utilizado não continha nenhuma informação sobre as equações subjacentes.

Há atualmente questionamentos sobre até que ponto o uso intensivo de ‘robôs

oráculos' levará à perda do interesse dos pesquisadores em descrever fenômenos em termos de equações e princípios unificadores. Esse debate é sintetizado pela confronto das noções de “predição” e “entendimento”.

??) expõe os elementos desse debate com um exemplo histórico. Como destaca, durante mais de um milênio, o movimento dos planetas eram descritos a partir de um modelo elaborado por Ptolomeu, cujos métodos lançavam mão de cálculos misteriosos envolvendo sobreposição de círculos. Apesar de ignorar a teoria da gravidade e de ter a Terra no centro do Universo, essa representação foi extremamente eficiente na sua capacidade preditiva. Ao mesmo tempo, pode-se dizer que ela era incompleta na medida em que não oferecia nenhum entendimento capaz de explicar o seu funcionamento.

Figura 5 – Modelo geocêntrico de Ptolomeu



Fonte: ??)

A descrição do movimento dos corpos celestes só foi finalmente compreendida a partir da equações diferenciais descobertas por Isaac Newton. Com elas tem sido possível, desde então, prever a trajetória de todo e qualquer planeta do sistema solar.

A qualidade da proposta de Newton estava na possibilidade de oferecer não apenas a descrição de trajetórias, mas por permitir também que entendamos o porquê que elas são de uma e não de outra forma. Ao nos trazer equações, ele nos facultou a compreensão do fenômeno do movimento a partir da ótica de princípios unificadores. E é exatamente nesse ponto que reside o poder da descrição matemática. Como analisa ??), se formos

capazes de extrair de um fenômeno complicado dois ou três princípios, podemos dizer então que o compreendemos.

Porém, dada a dominância de fenômenos complexos na natureza, tem sido extremamente difícil extrair de muitos deles princípios simples. Descrivê-los, portanto, a partir da obtenção de equações universalmente válidas pode ser uma forma um tanto ineficiente de gerar previsões relevantes (??). O emprego de técnicas de *machine learning*, nesse sentido, tem sido essencial.

A inteligência artificial tem ocupados outros espaços além da pesquisa científica. Desde robôs que leem a web para tomar decisão de investimentos (??) e carros que se auto dirigem, até contextos relativamente mais simples como tradutores automáticos e mecanismo buscas. Já no mundo do trabalho, um sem-número de projeções apontam para a substituição progressiva de quantidade considerável de atividades laborais por robôs.



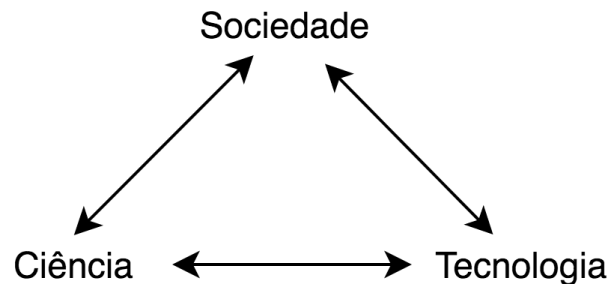
Figura 6 – Carro autônomo da Waymo, subsidiária de veículos da Alphabet (matriz do Google)

A produção estratosférica de dados e a invasão da inteligência artificial nos meios científicos e na indústria é sintomático da interação dinâmica entre a ciência, a tecnologia, e a sociedade em ciclos que se retroalimentam.

Novas descobertas levam a novas perguntas que exigem mais coleta de dados. Quando analisadas, essas informações são utilizadas para refinar e ajustar modelos, levando a novas perguntas e mais produção de dados. Esse processo tende a culminar na criação de novas tecnologias. Tecnologias, por seu turno, inspiram usos sociais criativos que quase sempre criam demandas por mais descobertas científicas e mais tecnologia. Sob esse

aspecto, podemos dizer que esses três atores desempenham o papel de forças motrizes da computação (??).

Figura 7 – Forças motrizes da computação



Adaptado de ??)

O barateamento dos custos de produção científica é outro aspecto relevante dessas transformações. À medida que a internet e dados, conseqüentemente, se tornam mais acessíveis, qualquer pessoa com boas ideias e bons hábitos de trabalho, de qualquer lugar, pode fazer ciência de primeira linha, comunicar seus resultados e aprender com a comunidade científica. Essa possibilidade é especialmente benéfica para países e instituições que não dispõem de instalações de pesquisa sofisticadas (??).

Sobre o papel da ciência da computação no desenvolvimento científico, ??) faz um consideração provocativa:

[...] a ciência da computação aplicada está desempenhando o papel que a matemática fez do século XVII ao século XX: fornecer uma estrutura ordenada e formal e um aparato exploratório para outras ciências. Além de seu aparentemente feliz caso com a teoria das cordas, é difícil dizer o que a matemática está fazendo para outras ciências hoje. A maioria dos cientistas de matemática que utilizamos hoje foi desenvolvida há mais de um século. (??, p. 131. Tradução nossa)

E é do do seio dessa reflexão que nasce a noção de um “pensamento computacional”. Tema que desdobraremos a seguir.

3 Pensamento computacional

A noção de pensamento computacional tem ganhado força desde 2006, ano em que a professora Jeannette Wing, então professora da Universidade Carnegie Mellon, publica um artigo seminal no qual propõe um conjunto de atitudes, ou abordagens, para a resolução de problemas fundamentadas na forma como cientistas da computação e programadores tratam problemas computacionais.

A definição clássica que muitos autores dão para o conceito é extraído do próprio trabalho da autora que estabelece que pensamento computacional consiste “numa abordagem para a solução de problemas, desenho de sistemas e entendimento do comportamento humano que se vale de conceitos fundamentais para ciência da computação” (??, Tradução nossa), ou ainda, na capacidade de formular problemas e expressar suas soluções de forma suficientemente clara de tal modo que um computador possa executá-las.

O desenvolvimento do tema requer algumas discussões preliminares. Começemos com uma apresentação sobre o papel dos algoritmos.

3.1 Algoritmos

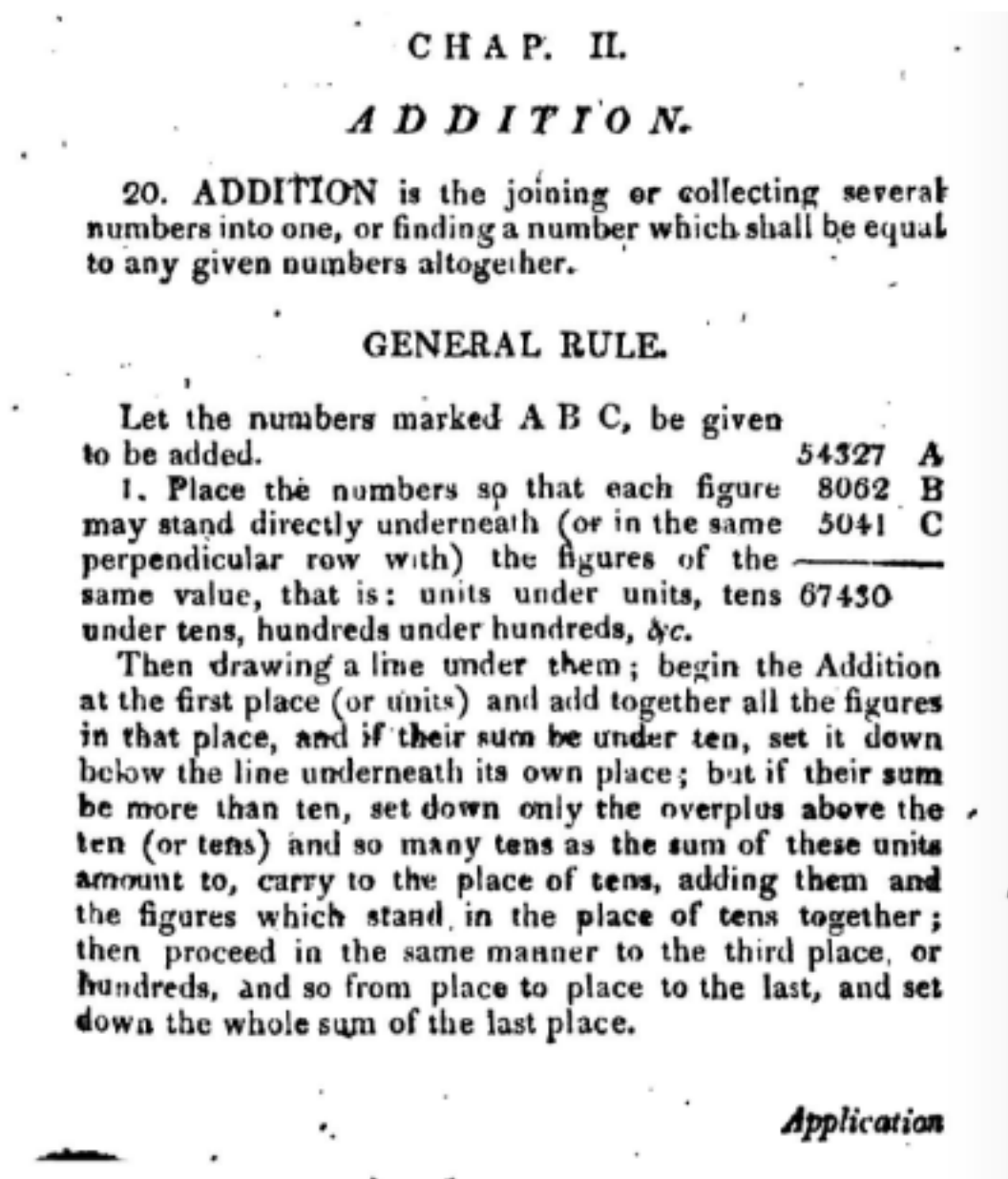
Em termos simplificados, um algoritmo nada mais é do que uma solução para um problema que satisfaz as seguintes condições (??):

- deve apresentar uma lista sequenciada de passos que levam à solução do problema
- deve ser um processo finito
- deve resolver qualquer instância do problema em questão

Na figura ?? lê-se um conjunto de instruções para somar três números inteiros, extraídos do livro *Practical Arithmetick in Four Books* do autor John Gough, publicado pela primeira vez em 1767. Sob o título ‘General rules’, lemos

Coloque os números de modo que cada algarismo possa ficar diretamente abaixo (ou na mesma linha perpendicular) dos algarismos de mesmo valor, ou seja: unidades em unidades, dezenas em dezenas, centenas em centenas.[...]Em seguida, desenhando uma linha abaixo deles; comece a adição no primeiro lugar (ou unidades) e some todas os algarismos naquele lugar, e se a soma delas for menor que dez, coloque-a abaixo da linha abaixo de seu próprio lugar;mas se a soma for superior a dez, estabeleça apenas o excedente acima das dez (ou dezenas) e algumas dezenas, à medida que a soma dessas unidades se eleva, carregue para o local das dezenas, adicionando-as e os algarismos que estão no lugar de

Figura 8 – Algoritmo de soma de três números inteiros



Fonte: ??)

dezenas juntos; em seguida, proceda da mesma maneira até o terceiro lugar, ou centenas, e de um lugar para outro até o último, e estabeleça a soma total do último lugar.

Perceba como essas instruções satisfazem as condições elencadas anteriormente: nela vemos um problema (cálculo da soma de três números inteiros) sendo resolvido de forma encadeada ao longo de um processo finito. Nota-se também que os passos acima facultariam a soma de quaisquer outros três números inteiros, atendendo a exigência de “resolver qualquer instância do problema”.

A tarefa de definir as instâncias de um problema equivale a determinar para quais perguntas um algoritmo deve oferecer respostas. Em problema de divisão, por exemplo, onde é preciso calcular a razão entre a e b , o par $a = 1.5$ e $b = 9.365$ representa uma delas. Já $a = 65.4$ e $b = 77$ compõe outra. Numa única sentença podemos resumir que qualquer par de reais, onde $b \neq 0$, representa uma instância válida.

A decisão de usar um computador para a resolução de qualquer problema exige que definamos tais instâncias. Sob o ponto de vista da ciência computação, ao procedermos dessa forma, distinguindo claramente os contornos do problema, definimos um “problema computacional”.

A definição clara das instâncias do problema (criação de um problema computacional) e dos passos exatos para sua solução¹ (elaboração de um algoritmo) nos permitirá automatizar a sua execução. E é exatamente nessas duas habilidades que se apoiam o pensamento computacional.

3.2 Abstração

A noção de abstração é outro componente que necessita ser compreendido ao tratarmos do assunto. Nas palavras de ??),

A abstração é usada na definição de padrões, generalização de instâncias e parametrização. É usado para deixar um objeto representar muitos. Ele é usado para capturar propriedades essenciais comuns a um conjunto de objetos enquanto oculta distinções irrelevantes entre eles (??, p. 1. Tradução nossa)

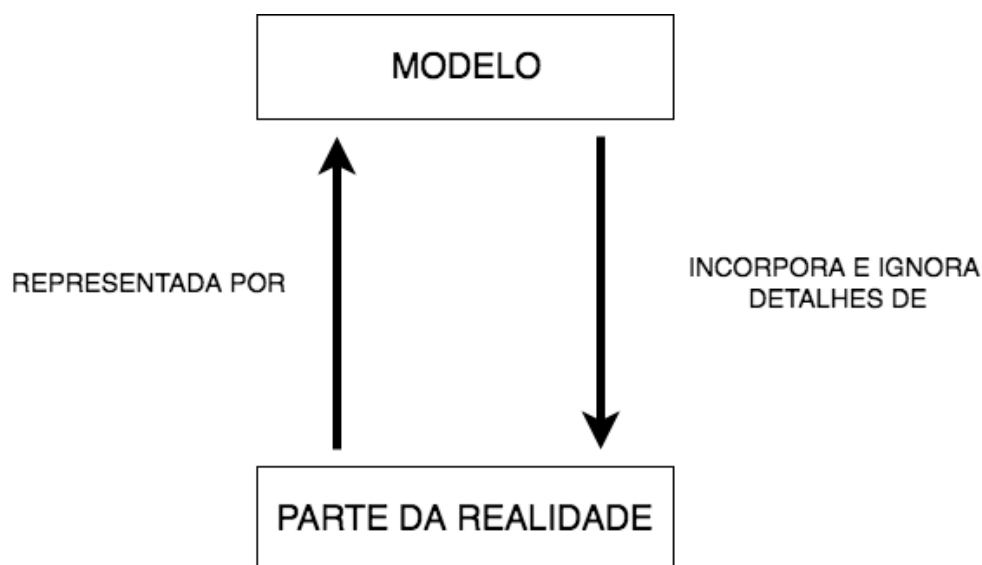
Abstrair, como enfatiza o trecho acima, equivale simplesmente a conduzir um processo de generalização, onde incorporamos parte dos detalhes da realidade em um modelo, ao mesmo tempo que descartamos outros. Estabelece-se assim uma relação entre dois níveis: a realidade e sua representação.

A figura ?? ilustra essa discussão. Nela vemos a obra “A Traição das Imagens” por René Magritte (1898-1967), onde lê-se “*Ceci n’est pas une pipe*” (“Isto não é um cachimbo”). Temos aí provocação que evidencia que um modelo nada mais é do que uma representação, e não a realidade em si.

O grau de detalhamento do modelo depende das circunstâncias do problema, bem como das necessidades de quem modela. Por exemplo, ao analisarmos a trajetória de lançamento de um foguete, podemos descartar suas dimensões e eliminar possíveis efeitos aerodinâmicos. Essa abordagem é extremamente conveniente para o ensino de primeiras

¹ Ao tratarmos computacionalmente um problema devemos ser capazes também de distinguir quando e porque, eventualmente, ele não tem solução.

Figura 9 – Relação entre a realidade e o seu modelo



Adaptado de ??)

Figura 10 – “A Traição das Imagens” por René Magritte, 1929

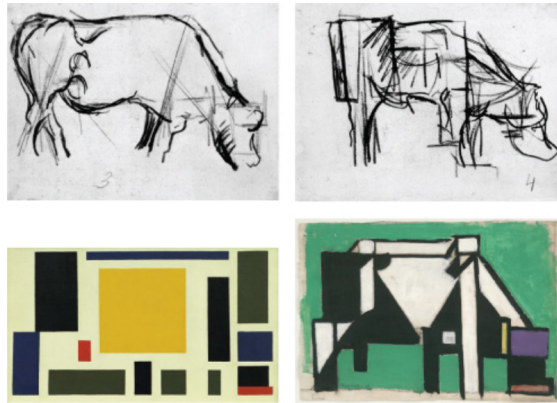


noções de física básica, mas demasiadamente simplória no contexto da condução de um programa espacial.

Na figura ?? temos a ilustração dessa ideia: a mesma “realidade” vaca representada por quatro modelos com diferentes graus de detalhes incorporados.

É interessante notar a proximidade do significado para abstração discutido até aqui com aquele proposto pelo filósofo John Locke, segundo o qual formação de uma abstração corresponde a uma transformação durante a qual “ideias tomadas de seres particulares se tornam representantes gerais de todos do mesmo tipo” (??, ?? apud ??, ??, p. 354.

Figura 11 – Quatro representações de uma vaca por Theo van Doesburg, 1917–1918



Fonte: ??)

Tradução nossa).

3.3 Modelos e módulos: duas possibilidades de abstração

??) distingue dois tipos de abstrações: a “abstração como modelo”, onde detalhes da realidade observada são desconsideradas em favor de outras – o mesmo trabalhado até aqui – e a “abstração como encapsulamento”, processo no qual organizamos nosso modelo em módulos ou cápsulas que permitem a composição de sistemas mais complexos.

Para diferenciarmos esses dois tipos tomemos como ponto de partida o planetário mecânico ilustrado na figura ???. Nele vemos uma simulação do sistema solar, um modelo, que como tal ignora certos aspectos da realidade. Neste caso em específico temos a desconsideração típica de representações astronômicas onde as proporções entre as dimensões dos planetas e a distância relativas entre eles é extremamente grande. A construção de modelos, ignorando e incorporando detalhes, é o que ??) chama de “abstração como modelo”.

O mesmo dispositivo incorpora a noção do que o autor chama de “abstração como encapsulamento”. Na figura ?? vemos um sistema de rodas dentadas típico do que podemos encontrar no interior de um planetário mecânico. São elas que permitem o movimento das esferas que representam os planetas. A superfície metálica que vemos na figura ?? cumpre o papel de interface do modelo ao escondê-las, deixando externamente visível apenas o que é relevante: o movimento de translação e rotação.

Distingue-se assim a formação de duas camadas durante o processo de encapsulamento: a interface com a qual se pode interagir com o modelo (o invólucro metálico do planetário) e a sua implementação propriamente dita (sistema de engrenagens encerradas pela interface). A figura ?? ilustra a relação entre elas.

Figura 12 – Planetário mecânico



Fonte: ??)

Figura 13 – Engrenagem de um planetário mecânico



Fonte: ??)

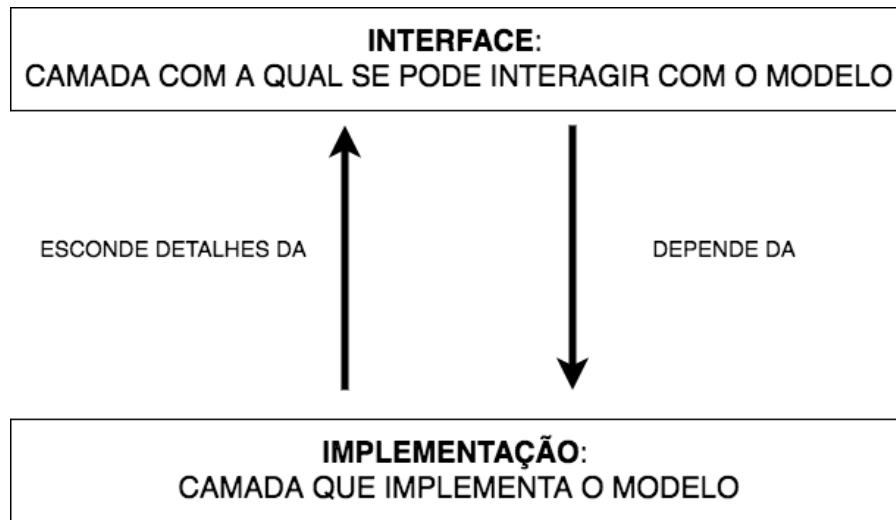
O uso do encapsulamento é essencial na ciência da computação. Nesse contexto, essa estratégia toma forma em funções e estrutura de dados². Tomemos um exemplo envolvendo funções. Considere a necessidade de calcularmos a seguinte expressão:

$$\frac{(2 + 3 + 5 \times 7)}{5}$$

Usando uma linguagem de programação poderíamos reescrever a equação acima do seguinte modo:

² Por estrutura de dados nos referimos às diversas possibilidades de organização e relacionamento de dados que uma linguagem oferece. Dentre as mais comuns estão os vetores, as listas, as pilhas...

Figura 14 – Interface e implementação - dois aspectos de um modelo



Fonte: ??)

```

razao(soma(2, 3, producto(5, 7)), 5)

```

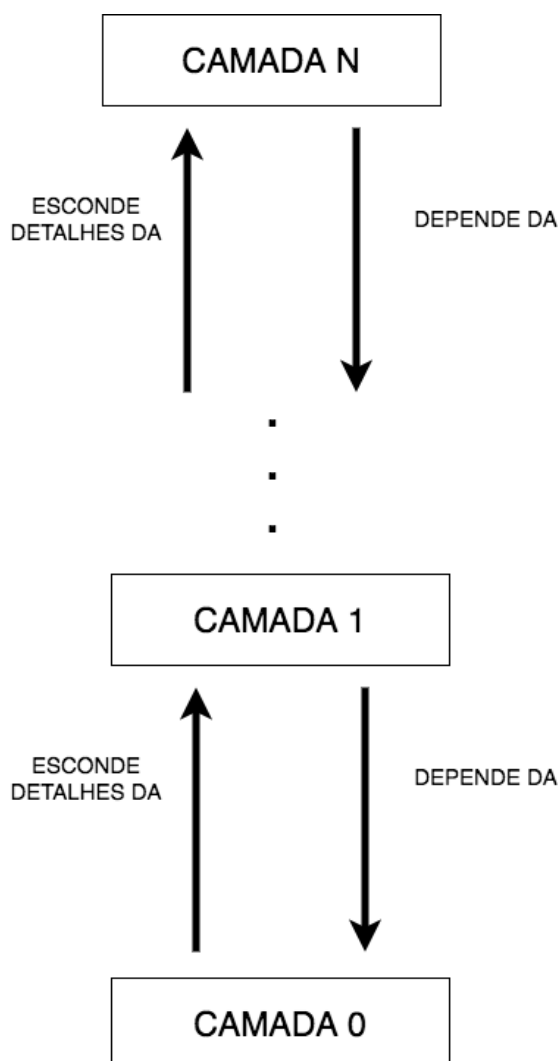
Perceba a formação de cápsulas. Por exemplo, aos escrevermos *produto*(*a*, *b*, *c*...), estamos modularizando o processo de multiplicação, pois todos os detalhes dessa operação estão invisibilizados atrás de uma interface, cujo nome decidimos chamar como “produto”. Aplica-se o mesmo raciocínio aos casos das funções *soma*(*a*, *b*, *c*...) e *razao*(*a*, *b*).

A decomposição do modelo ou sistema pode ganhar tantas unidades quanto se queira, até atingirmos o nível binário (0 e 1s). É exatamente na possibilidade de aplicação recursiva dessa estratégia que a construção de sistemas complexos se tornam viáveis. Esse aspecto é ilustrado na figura ??.

De fato, ao encapsular, programadores buscam expressar-se apenas em termos de unidades representativas. Essa tática permite a eles raciocinar somente em termos do problema que estão resolvendo, e os protege assim das complexidades próprias do ambiente computacional com que estão trabalhando, tal como a necessidade de administrar diretamente o uso da memória, por exemplo. Ao mesmo tempo, por favorecer a organização e a estruturação lógica do código fonte, como vantagem adicional essa abordagem facilita manutenções futuras do sistema.

Todo o desenvolvimento da computação está assentado na decomposição modular de sistemas. Celulares, computadores, sistemas operacionais, automóveis... Toda e qualquer tecnologia operada por dispositivos eletrônicos. Há exemplos históricos, contudo, onde esse princípio é aplicado apesar de não haver nenhum substrato eletrônico. Na figura ?? vemos

Figura 15 – Decomposição em camadas de um sistema

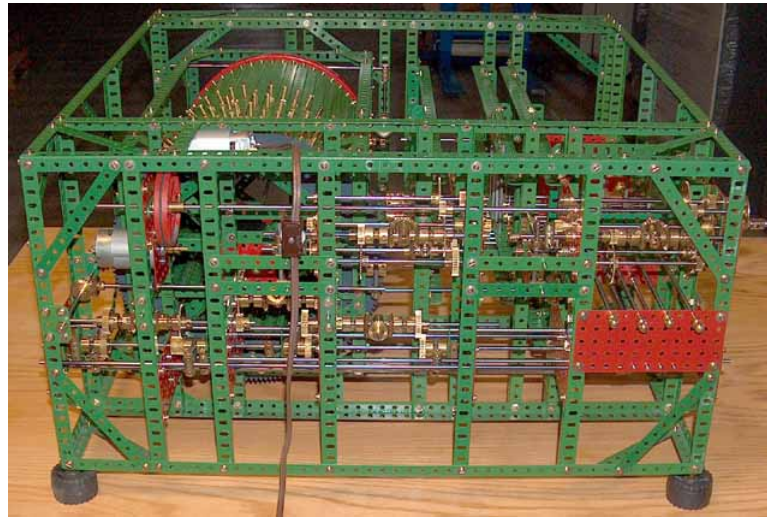


Fonte: ??)

o primeiro computador programável desenvolvido com o propósito de executar operações matemáticas. Conhecido como Meccano, ele foi proposto por Charles Babbage (1791–1871). Movido a vapor, na imagem temos um exemplar construído em latão e ferro.

Em suma, a criação de abstrações tanto por modelagem quanto por encapsulamento nos permitem administrar as complexidades do mundo real. Criando modelos, buscamos diminuir os aspectos ruidosos do mundo real em favor de elementos relevantes para a análise do problema, enquanto que ao encapsularmos, como diz ??), evitamos ser vitimizados pelos intrincados detalhes do “mundo dos computadores”.

Figura 16 – Meccano – Implementação do primeiro computador programável proposto por Charles Babbage



Fonte: ??)

3.4 Um apanhado geral

Até agora apresentamos alguns conceitos sem relacioná-los apropriadamente. Alguns deles:

- modelo matemático
- problema computacional
- algoritmo
- abstração como modelo e encapsulamento

Afinal como esses elementos se articulam em torno do conceito de pensamento computacional, tema-central desse trabalho? Nessa seção, essa pergunta será o nosso fio-condutor.

Começemos relacionando os conceitos de modelo matemático e problema computacional.

O propósito da construção de modelos matemáticos se confunde com o da atividade científica: entender e descrever fenômenos e o comportamento de sistemas em função de condições pré-estabelecidas.

Na construção de tais modelos, busca-se investigar as regras de interação entre os componentes internos ao sistema investigado e como o ambiente externo influencia essas

relações. Fixados alguns parâmetros, tenta-se entender qual é o papel da variação de alguns outros. Em muitos casos, o objetivo principal desse empreendimento é a formalização dessa descrição em equações matemáticas. Em outros, a validação de descrições já propostas.

O surgimento do modelo matemático nasce portanto da consolidação das equações que regem o sistema.

Tanto na engenharia quanto nas ciências naturais, o uso de modelos matemáticos tem visado a construção de simulação computacionais. A adoção dessa estratégia favorece o entendimento do comportamento do sistema sob condições limites o que pode evitar, em alguns casos, a ocorrência de tragédias. Ao mesmo tempo, por oferecer um contexto que proporciona o ganho de *insights*, o uso de simulação facilita o desenho de estratégias para otimização de recursos materiais e humanos.

A figura ?? torna explícita a proximidade entre a construção de um modelo matemático e a resolução de um problema computacional. Como definido na seção ??, entende-se como problema computacional um conjunto de perguntas que um computador potencialmente responder. A tarefa de resolver o problema computacional, como visto, se resumirá a construção de um algoritmo que leve a cada uma dessas respostas. Igualmente, as equações que compõe um modelo matemático tem como papel tecer uma associação entre um conjunto de parâmetros e uma configuração específica assumida pelo sistema sob observação. Nesse sentido, relações matemáticas e algoritmos cumprem a mesma finalidade.

Essa forte correlação estrutural é o que tem tornado efetiva a relação entre a ciência da computação e a pesquisa científica. Os elos que integram esses domínios estão visíveis na figura ??.

Figura 17 – Elementos de um modelos matemático

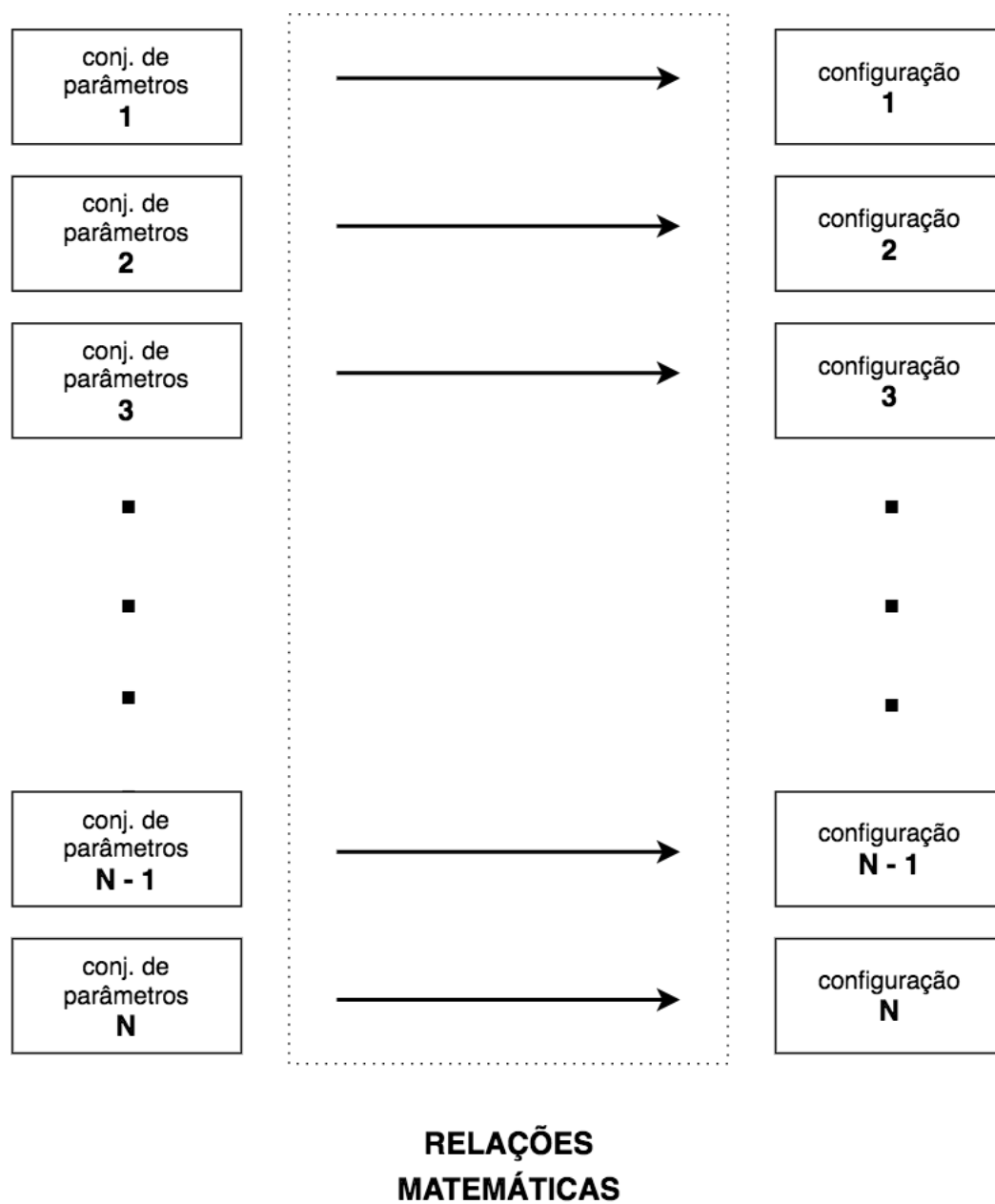
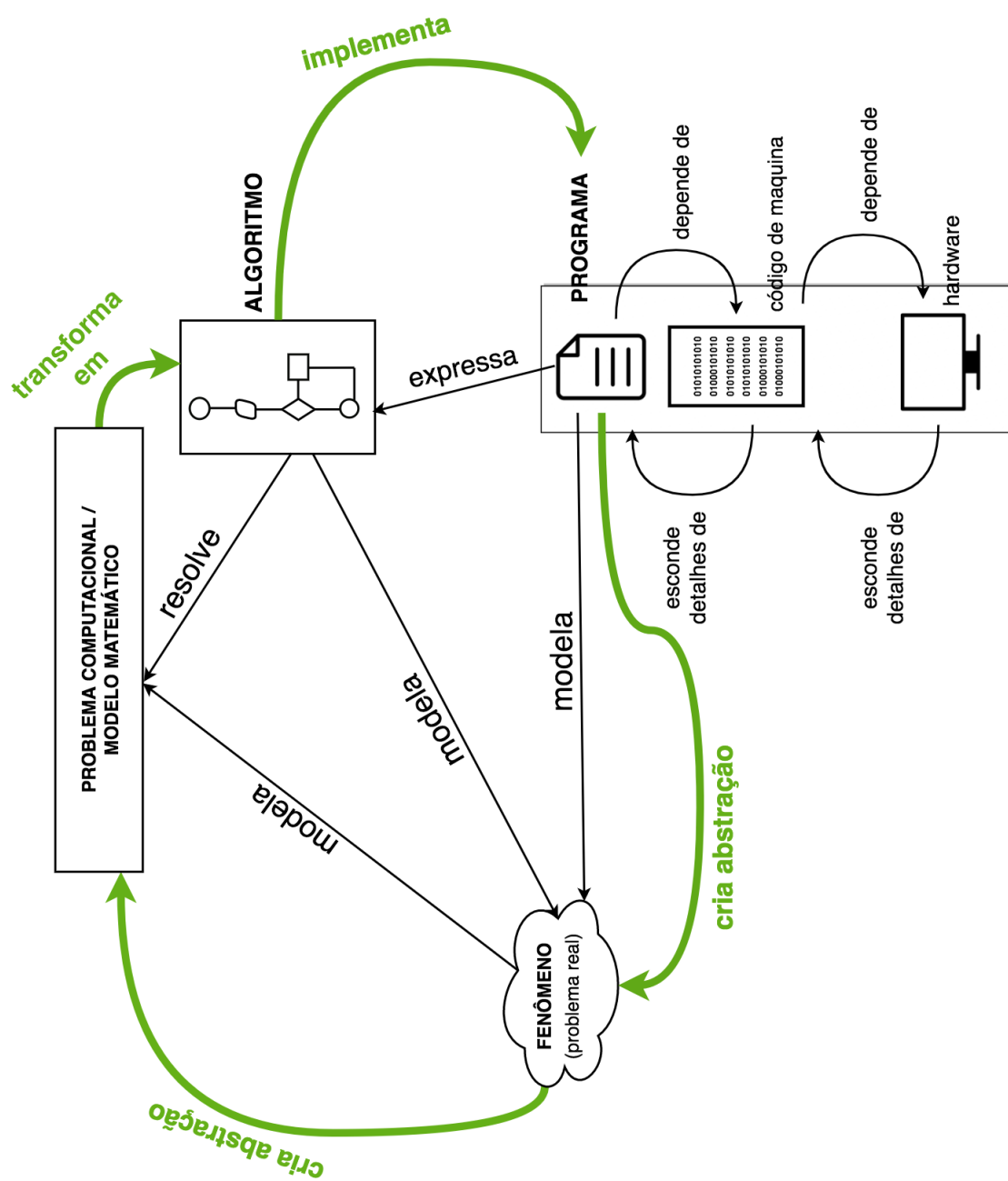


Figura 18 – Visão panorâmica do pensamento computacional



4 Implicações para aprendizagem científica

O desenvolvimento da computação e o amadurecimento da noção de pensamento computacional traz consigo várias implicações e oportunidades educacionais.

De partida, pode-se dizer que a efetivação do uso do computador como ferramenta para resolução de problemas exige a formação de capital humano. Normalmente, espera-se que estudantes tenham acesso à universidade para introdução das primeiras noções de programação e ciência da computação. Sabe-se contudo que eles se mostram cada vez familiarizados com *smartphones* e outros dispositivo computacionais. Por isso podemos nos perguntar: por que não antecipar essa instrumentalização?

Esse questionamento se torna ainda mais pertinente em face do percentual reduzido de portadores de grau universitário. A demanda por profissionais com experiência na resolução de problemas computacionais tem crescido numa proporção superior a que academia tem formado. Esse desequilíbrio tem levado grandes empresas de tecnologia como o Google e IBM a dispensar a formação universitária como pre-requisito para contratação (??).

A motivação para essa antecipação não é apenas de caráter laboral. Como discutiremos ao longo desse capítulo o uso de computadores e do pensamento computacional oferece excelentes contextos para desenvolvimento de competências científicas. O reverso também é verdadeiro: problemas científicos e matemáticos são domínios excelentes para a aplicação e exercício do pensamento computacional. É nessa tese que esse trabalho se apoia.

No capítulo ?? mostramos o como conceitos de abstração, decomposição de problemas e simulação, aliados a praticas de representação de problemas se estruturam em torno do pensamento computacional. Ao mesmo tempo que são centrais na ciência do computação, estes elementos também são fundamentais para o desenvolvimento de modelos e para a compreensão e resolução de problemas num largo espectro de disciplinas matemáticas e científicas (??).

A literatura tem demonstrado de diversas formas o como a aprendizagem de programação - uma das praticas de representação - em conjunto com conceitos de outros domínios pode ser mais fácil do que aprender cada um desses tópicos individualmente. Essa abordagem é destacadamente diferente da forma como o ensino de computação tem sido trabalhado no ambiente escolar, onde as atividades propostas focam em aspectos apartados da realidade.

Em um estudo citado por ??), descobriu-se que a introdução de programação no

contexto da modelagem de fenômenos físicos e químicos, durante as atividades de alunos de graduação, resultaram no aprendizado mais efetivo de programação, e no aumento do engajamento com a área de domínio das tarefas.

A literatura também nos apresenta outros exemplos de contextualização de atividades computacionais com o ensino de ciências. Dentre algumas propostas práticas, podemos destacar o uso eficiente do software Netlogo (??) na introdução de noções de probabilidade e estatística, como relatado por ??), e na modelagem de fenômenos epidemiológicos, tal como reportado em (??).

Vários estudos apontam que o enriquecimento trazido por essa abordagem conjunta, associada com o caráter “mão na massa” de muitas dessas atividades, produz impactos positivos na forma como os alunos percebem as aulas de ciências (????). A sensação de estar resolvendo um problema autêntico com real aplicabilidade e, em muitos casos, inserido na realidade do qual fazem parte, produz aumentos sensíveis dos níveis de engajamento.

??) sugere que o aumento da atratividade dessas disciplinas, apresentadas dessa forma, também pode favorecer o aumento da representação de grupos minoritários nos meios científicos, tais como mulheres, negros e transgêneros.

A atenção crescente que educadores têm dado ao tema do pensamento computacional tem sido apoiado também pela expectativa de inverter a relação entre o estudante e a tecnologia. A pergunta que se coloca é: como de meros usuários eles podem vir a tornar criadores?

??) argumenta que apesar de serem vistos como “nativos digitais”¹, a geração nascida durante esse século está familiarizada apenas com o consumo de tecnologia, e não necessariamente com a sua criação ou produção. Para ele, o benefício de dotá-la com essa capacidade criativa não estará apenas nessa ou naquela tecnologia desenvolvida pelo estudante, mas sim nas competências cognitivas adquiridas por ele durante o processo de criação. Dentre algumas delas, o autor destaca:

1. a capacidade de pensar em termos sistêmicos, ou seja, em função do comportamento agregado resultante da interação dos componentes de sistema - “o todo não é a soma das partes”;
2. a habilidade para trabalhar em equipe;
3. a competência para estruturar e planejar as etapas de um projeto;
4. a facilidade para experimentar novas ideias (prototipagem);

¹ A expressão “nativo digital” foi cunhada pelo escritor Marc Prensky para designar aqueles que desde o seu nascimento tiveram a oportunidade de interagir com tecnologias digitais, tais como videogames, computadores, telefone celular, iPhones, iPads...

5. a destreza para decompor ideias complexas em unidades menores (abstrações);
6. a perícia necessária para investigar defeitos ou comportamentos inesperados (deputação);
7. a inteligência emocional necessária para lidar com frustrações e perseverar em face de dificuldades que surgem ao longo de um projeto.

Como ressalta, mesmo que o estudante não venha a se tornar um engenheiro ou um cientista da computação todas essas competências são importantes para todo e qualquer tipo de atividades. Analogamente, mesmo que poucos deles se profissionalizem como escritores, a capacidade para redigir textos é fundamental para qualquer um.

??) direciona o seu argumento para justificar os benefícios do ensino de programação. Veremos, contudo, que o desenvolvimento dessas competências criativas - todas elas associadas ao pensamento computacional - podem passar por caminhos diferentes da habilidade de escrever código, propriamente dita.

Outras oportunidades educacionais estão associadas às respostas que essa abordagem oferece para vários problemas relacionados ao tema da avaliação. O ensino de ciências é dominado por uma tradição explicativa cujo objetivo é a simples transmissão de dados e conceitos pelo professor. Nesse contexto, convencionou-se que o ciclo de ensino e aprendizagem se fecha quando os alunos são capazes de reproduzir essas informações numa prova cronometrada. No ensino de física, em especial, entende-se que o aprendizado se efetiva quando eles se mostram capazes de resolver três ou quatro variações de um problema discutido em sala de aula - quase sempre de natureza algébrica.

O que se observa, portanto, é um modelo avaliativo indutor de um comportamento autômato do estudante, em prejuízo do desenvolvimento da sua capacidade crítica e analítica.

Por outro lado, o desenvolvimento de atividades sustentadas pelo pensamento computacional tem como foco a resolução de problemas abertos, quase sempre relacionados à construção e não à imposição de modelos (abstrações). Conceitos e princípios científicos são trabalhados sob a perspectiva de que não há modelos “corretos”, mas sim “apropriados”, que melhor respondem a determinadas questões.

Dessa forma, a absorção de tais conteúdos conceituais se dá pela necessidade de articulá-los, por exemplo, na construção de uma animação, ou até mesmo de um jogo. Nesse ambiente, o aluno tem a oportunidade de atestar a validade de um princípio físico pela “qualidade” do caminhar do seu personagem, por quão bem o seu comportamento reflete a realidade. O aprendizado das três leis de Newton nesse cenário nascerá da análise, da observação, e não da simples “decoreba”.

Os marcos avaliativos introduzidos por essa abordagem estão ancorados na perspectiva de que a aprendizagem de conteúdos científicos devem ser encarados como um meio, um *playground* para o desenvolvimento de competências cognitivas, e não como um fim em si.

Por esse motivo pode-se afirmar que o uso do pensamento computacional estimula comportamentos em sala de aula mais compatíveis com a natureza da investigação científica. E por ela se desenvolver cada vez mais em bases computacionais, como discutido no capítulo ??, apoiar a sua aprendizagem na construção e na análise de modelos computacionais permite que os alunos tenham uma visão mais realista e coerente com a forma que ela é exercida profissionalmente.

Dadas as motivações para exercício do pensamento computacional, ainda permanecem algumas questões de natureza práticas a serem respondidas e conciliadas para que essa abordagem se viabilize em sala de aula. Mesmo que até aqui tenhamos esboçado algumas definições, elas ainda não foram capazes de oferecer respostas precisas sobre como esse conceito pode ser materializado em sala de aula. Essa é tarefa que desejamos cumprir no capítulo seguinte.

4.1 Significado do pensamento computacional no contexto da sala de aula

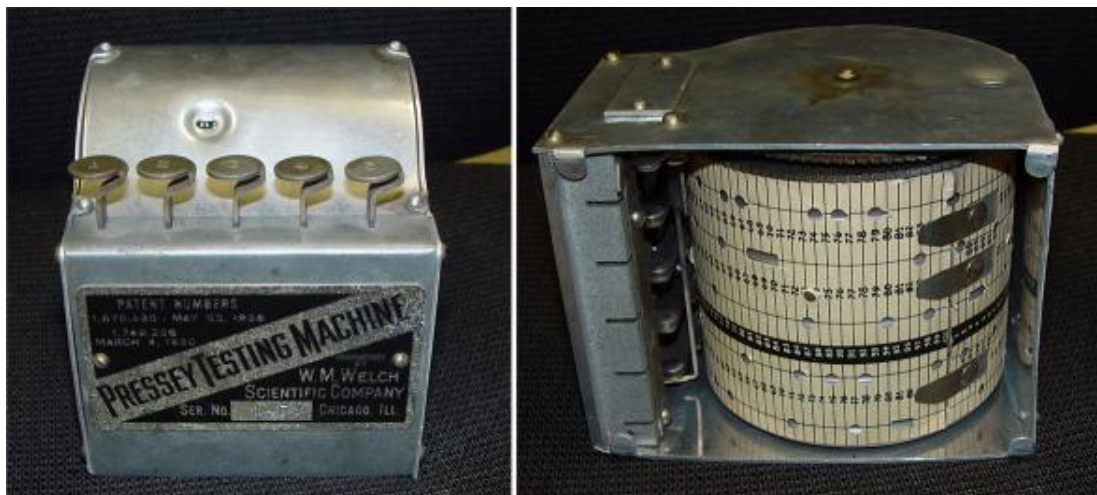
O emprego de dispositivos autômatos como ferramentas educacionais remonta às “máquinas de ensinar”, cujo movimento dependia da seleção da alternativa correta de uns dos itens de uma questão múltipla escolha. Esse mecanismo foi proposto e desenvolvido pelo professor de psicologia da Universidade de Ohio, Sidney L. Pressey.

O uso da expressão “pensamento computacional”, contudo, tem uma origem mais recente. O seu primeiro registro é observado no livro *Mindstorms: Children, computers and powerful ideas*, da década de 1980, onde Seymour Papert discorre sobre a oportunidade trazida por computadores para o ensino de matemática.

A popularização da expressão e o consequente interesse da comunidade acadêmica veio apenas com a publicação do artigo por ??), no qual é proposto “um conjunto de atitudes e habilidades universalmente aplicáveis” assentadas na forma como cientistas da computação e programadores resolvem problemas.

Apesar do intenso debate despertado pelo artigo e do reconhecimento das questões e oportunidades educacionais implicadas no tema, o desenvolvimento de atividades e abordagens para sala de aula tem sido especialmente desafiador em face da parcial ou completa inexistência de um significado consensual para o conceito que seja adequado para este ambiente. A definição para o pensamento computacional resultante da discussão

Figura 19 – “Máquina de ensinar” de Pressey



Fonte: ??)

proposta por ??) foca apenas na contribuição da ciência da computação para as várias áreas da atividade humana, e acaba por não demonstrar como esse conceito deve se realizar no plano educacional.

Essa indefinição tem tornado difícil o estabelecimento de formas mensuráveis de avaliar o progresso dos alunos e tem redundado na inviabilização do desenvolvimento de um currículo, e conseqüentemente, na oferta de treinamento de professores.

Na tentativa de preencher essa lacuna, um número significativo de conferências e congressos tem sido realizado ao redor do mundo, em especial nos Estados Unidos e na Europa, financiados em boa parte por secretarias de educação e grandes empresas de tecnologia, como a Microsoft. Ao mesmo tempo, desde da publicação de ??), uma quantidade expressiva de artigos reportando experimentações em sala de aula tem sido publicados.

O ponto de convergência desses debates tem sido a busca de uma definição clara sobre o significado desse conceito para o contexto da sala de aula que ao mesmo tempo dialogue com as particularidades e dificuldades pertinentes a esse ambiente.

Dentre as questões para as quais se tem buscado resposta incluem-se (??):

1. Como o pensamento computacional se relaciona com as atuais disciplinas e práticas correntes em sala de aula?
2. Como se estabeleceria a progressão de assuntos a serem trabalhados?
3. Quais são as práticas educacionais refletidas pelo pensamento computacional, e como preparar professores para que possam empregá-las em sala de aula?

4. Qual é melhor forma de avaliar o uso e o exercício dessas práticas pelos alunos?

Alguns exemplos concretos da formulação desses e de outros questionamentos podem ser encontrados no relatório produzido pelo Conselho Nacional de Pesquisas dos Estados Unidos (NRC), resultante de um encontro realizado no ano de 2010. Nele são listados vinte habilidades e práticas identificadas com o pensamento computacional, tais como a abstração e decomposição de problemas, formulação de soluções heurísticas e de estratégias de busca, bem como conceitos pertinentes à ciência da computação, tais como processamento paralelo e uso de recursão (??, ?? apud ??, ??).

Também vale registrar o trabalho de ??), no qual o esforço de significar o pensamento computacional na perspectiva da sala de aula traduziu-se no mapeamento de algumas competências a disciplinas comuns à educação básico. Esse conteúdo está disponível na tabela.

Tabela 4 – Níveis de investigação.

longtable llllll	Práticas do Pensamento Computacional	Ciência da Computação	Matemática	Ciências
Estudos sociais	Linguagens			
Coleta de dados	Encontre uma fonte de dados para uma área problemática	Find a data source fora problem area, forexample, fl ipping coinsor throwing dice	Collect data from anexperiment	Study battle statisticsor population data
Análises de dados	Escreva um programa de cálculo de cálculos básicos em um conjunto de dados	Count occurrences offl ips, dice throws andanalyzing results	Analyze data from anexperiment	Identify trends in datafrom statistics
Representação de dados	Use estruturas de dados como matriz, lista de links, pilha, fila, gráfico, tabela de hash etc. Use histogram, piechart, bar chart torepresent data; usesets, lists, graphs, etc.To contain data	Summarize data froman experiment	Summarize andrepresent trends	Represent patternsof different sentencetypes
Decomposição de problemas	Defi ne objects andmethods; defi ne mainand functions	Apply order ofoperations in anexpression	Do a speciesclassifi cation	Write an outline
Abstração	Use procedures toencapsulate a setof often repeatedcommands thatperform a function;use conditionals,loops, recursion, etc. Use variables inalgebra; identifiyessential facts in aword problem; studyfunctions in algebracompared to functionsin programming;Use iteration to solveword problems	Build a model of aphysical entity	Summarize facts;deduce conclusionsfrom facts	Use of simile andmetaphor; write astory with branches
Algoritmos e Procedimentos	Study classicalgorithms; implementan algorithm for aproblem area	Do long division,factoring; do carries inaddition or subtraction	Do an experimentalprocedure	Write instructions
Automação	Use tools such as:geometer sketch pad;star logo; python codesnippets	Use probeware	Use excel	Use a spell checker
Paralelização	Threading, pipelining,dividing up data ortask in such a wayto be processed inparallel	Solve linearsystems; do matrixmultiplication	Simultaneously runexperiments withdifferent parameters	
Simulação	Algorithm animation,parameter sweeping	Graph a function in aCartesian plane andmodify values of thevariables	Simulate movement ofthe solar system	Play age of empires;Oregon trail
	Do a re-enactmentfrom a story			

Fonte: ??)

Apesar da ausência de consenso, um conjunto numeroso de aplicações bem sucedidas tem apontado para algumas diretrizes. Apoiando-nos em algumas dessas orientações, na próxima seção iremos tentar delimitar quais as competências específicas relacionadas ao

pensamento computacional que podem ser trabalhadas pelos alunos e de que forma elas tendem a favorecer a tão almejada aprendizagem científica.

E como definições só podem ser úteis se forem acompanhada de exemplos práticos, iremos, em seguida, demonstrar como o exercício dessas competências podem ser incorporadas a temas das aulas de ciências, ao mesmo tempo que ilustramos a discussão com a descrição de casos reais.

4.2 Taxonomia

Na esteira do debate sobre como incluir o pensamento computacional nas aulas de ciências, um conjunto de soluções vem sendo encaminhados.

O problema