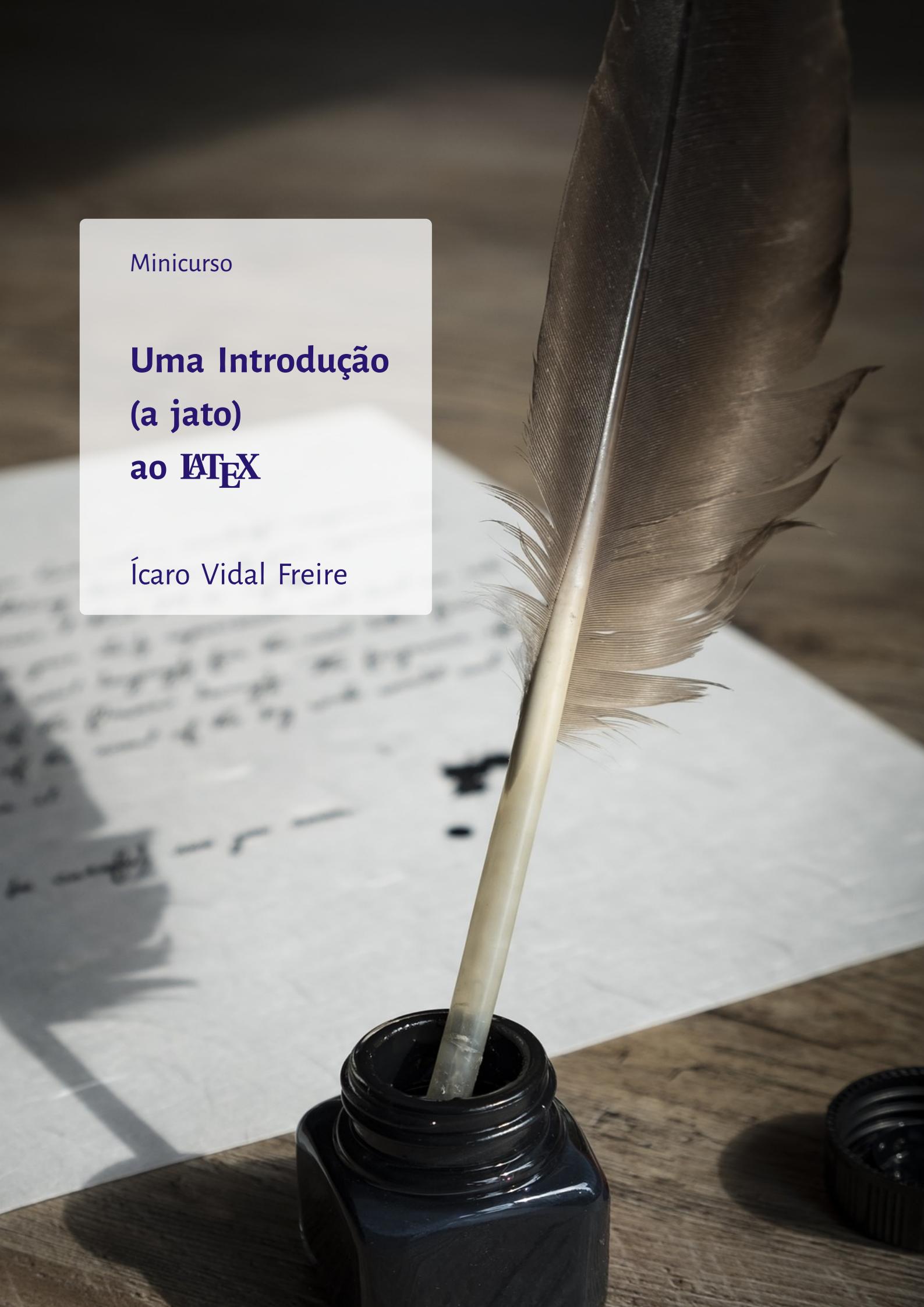


Minicurso

Uma Introdução (a jato) ao L^AT_EX

Ícaro Vidal Freire



Sumário

I A Arte de Nomear as Coisas	4
1 A Seiva da Árvore ...	4
1.1 Diferenciando os nomes	4
1.2 O surgimento do L ^A T _E X	5
1.3 Mais e mais nomes: compiladores, interpretadores e formatos	5
1.4 Como colocar o L ^A T _E X em meu computador: as Distribuições	8
2 Sim ...mas como eu uso?	8
2.1 Como fazemos a "compilação"?	10
2.2 Editores para L ^A T _E X	11
2.2.1 O L ^A T _E X é outro Word?	11
2.2.2 Indicações de editores para L ^A T _E X	11

Bate-papo Inicial

Se você optou por ler esse material, vá até o fim! O L^AT_EX pode ser para você uma ferramenta extremamente agradável para produção tipográfica, a nível profissional, de suas futuras notas de aula; monografias, dissertações ou teses; listas de atividades; partituras; marcações no xadrez; etc.



Não pretendo abordar muita coisa, confesso.

Fui incumbido de ministrar, em pouco mais de 3h, uma introdução ao L^AT_EX.



Exemplo de preâmbulo

```
1 % arara: lualatex: {options: ['--output-directory=docs_saida']}
2
3 % classe do documento -----
4 \documentclass[12pt]{article}
5
6 % pacotes carregados -----
7 \usepackage{mathtools, amsfonts, amsthm}
8
9 % configurações do título -----
10 \title{\textbf{Título}}
11 \author{Ícaro Vidal Freire}
12 \date{\today}
13
14 =====
15 % Início do Documento
16 %
17 \begin{document}
18 %
19 \maketitle % produz a saída do título
20 %
21 \begin{abstract}
22 Pequeno resumo.
23 \end{abstract}
24 %
25 \section{Primeira Seção}
26
27 Olá \LaTeX!
28 %
29 \end{document}
30 =====
```

A Arte de Nomear as Coisas

1 A Seiva da Árvore ...

Lembro-me da primeira vez que vi o nome \LaTeX ...

Foi em uma chamada de um minicurso de alguma "semana de matemática" da universidade que fiz graduação. O título era algo assim: "Introdução ao \LaTeX ".

Eu estava a um semestre de concluir a graduação e pensei: *"Rapaz... acho que eles erraram esse minicurso. Pra quê estudar látex em Matemática? Isso está mais para Geografia."*

Sim! Eu pensei que estavam falando daquela substância espessa e branca que sai de algumas plantas (seringueira, por exemplo)!

Então, vamos deixar as coisas claras: não estamos falando de látex, mas de \LaTeX .

Como pronunciar corretamente?

Aliás, como se pronuncia a palavra \LaTeX ?

Existem, pelo menos, duas maneiras de pronunciarmos, corretamente, essa palavra: "*LeiTéc*" ou "*LaTéc*", ou seja, o som de $\text{T}_{\text{E}}\text{X}$ ("Téc") é o mesmo que na palavra "*tecnologia*". Particularmente, adoto a segunda opção. Evite, por amor a Deus, falar "*Látecks*".

Falando nisso, essa palavra $\text{T}_{\text{E}}\text{X}$ não está destacada de forma aleatória! Ela foi idealizada como sendo a junção de três letras gregas: $\tau \epsilon \chi$.¹ Esse núcleo grego gera palavras como *arte* ($\tau \epsilon \chi \nu \eta$) ou mesmo *tecnologia* ($\tau \epsilon \chi \nu \omega \lambda \sigma \gamma \iota \alpha$). Daí vem o espírito da palavra \LaTeX : unir uma arte (a tipografia) com a tecnologia (programação) para produzir documentos com uma beleza realmente ímpar.

Na realidade, para falarmos com propriedade sobre \LaTeX , precisamos tangenciar o $\text{T}_{\text{E}}\text{X}$.

1.1 Diferenciando os nomes

Tudo começou quando **Donald Ervin Knuth** queria qualidade nos elementos tipográficos de seus livros, principalmente na escrita matemática. Ele então criou, em meados da década dos anos 70, um sofisticado programa para a composição tipográfica de textos científicos e uma alternativa quase necessária para

Seção 1. A seiva da árvore
Seção 1. A seiva da árvore
Seção 1. A seiva da árvore



Figura 1. Isso é látex, não \LaTeX

edição de textos com conteúdo matemático. Nasceu o \TeX . Em suas próprias palavras:

\TeX é destinado para a criação de belos livros - e, especialmente, para os livros que contêm grande quantidade de matemática.

Todavia, ao que parece, essa linguagem de programação não era tão acessível a meros mortais como nós, por conter diversos parâmetros relativos ao formato final do texto ... Bom, vamos falar a verdade: dava muito trabalho para digitar o que você queria simplificar. Isso porque o \TeX era formado por objetos denominados "primitivos" e toda estruturação do texto deveria ser feita por meio deles. O próprio Knuth criou um conjunto de macros, ou seja, um mapeamento de sequências de comandos primitivos, frequentemente utilizados, para simplificar a escrita de comandos \TeX em seus livros. A esse conjunto de macros ele deu o nome de *Plain \TeX* .

Obviamente, o *Plain \TeX* é um conjunto de macros bem simples e que poderia ser expandido. E foi isso que aconteceu.

1.2 O surgimento do \LaTeX

Leslie B. Lamport, facilitou nossa vida! Por volta dos anos 80, ele criou um conjunto de macros para o \TeX , muito bem estruturado, e com ideias interessantes (classes e pacotes, por exemplo) que somaram substancialmente à causa do \TeX , formando assim o \LaTeX . Inclusive, o "La", de \LaTeX ; vem do "La", de *Lamport*.²

Saiba mais

Uma *macro* (abreviação para macroinstrução), em ciência da computação, é uma regra ou padrão que especifica como uma certa sequência de entrada deve ser mapeada para uma substituição de sequência de saída de acordo com um procedimento definido.

² $\text{\LaTeX} = \text{Lamport} + \text{\TeX}$

Logo, nunca se esqueça disso:

O \LaTeX veio para facilitar sua vida!

1.3 Mais e mais nomes: compiladores, interpretadores e formatos

Bom ... já sabemos que \TeX e \LaTeX são coisas diferentes, entretanto, inter-relacionadas: o segundo é a forma mais utilizada, atualmente, para interagir com o primeiro.

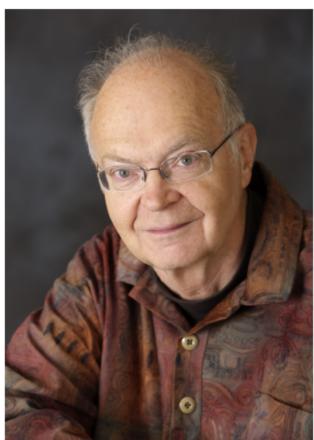
Nesse ponto, seria interessante diferenciarmos *engines* (motores/interpretadores) de *formats* (formatos).

O \TeX é um interpretador (*engine*); o \LaTeX , um formato (*format*).

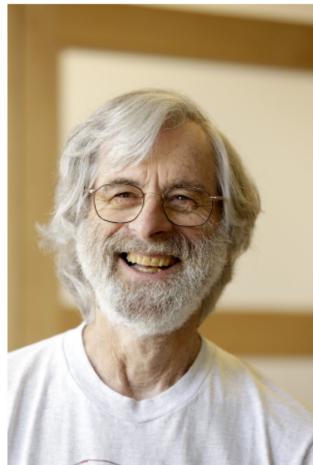
Os *interpretadores* são os arquivos binários executáveis (ou seja, o programa em si); já os *formats* são macros (comandos ou instruções), baseadas em \TeX , que usamos para escrever nossos documentos (a grosso modo, são linguagens ou atalhos para sequências de comandos ou estruturas em \TeX).

Saiba mais

Um outro formato que vimos é o *Plain \TeX* ; e, atualmente, um formato que se destaca é o *ConTeXt*.



(a) Donald E. Knuth



(b) Leslie B. Lamport

Figura 2. Os culpados!

Não é difícil perceber que o interpretador \TeX é bem antigo e que outros tenham surgido ao longo desses anos. De fato, à época do \TeX , nem existia ainda arquivos com extensão `.pdf` – a saída dos documentos era em DVI. Quando surgiu o PDF, um interpretador ficou bem conhecido: $\text{pdf}\text{\TeX}$ – cuja saída poderia ser em DVI que poderia ser convertida em PDF. O $\text{pdf}\text{\TeX}$ talvez seja o mais usado dentre os interpretadores que vamos falar, pois já vem selecionado, por padrão, em muitos editores dedicados ao \TeX ³; as pessoas, simplesmente, não mudam a configuração padrão se realmente não for necessário. Aliás, nosso sistema relaciona o \TeX com o $\text{pdf}\text{\TeX}$ usando um "atalho", denominado $\text{pdf}\text{\TeX}$.

Portanto, o $\text{pdf}\text{\TeX}$ (que geralmente é o nome que aparecerá em nossos editores) é o *interpretador* $\text{pdf}\text{\TeX}$ com o *formato* \TeX .

Atualmente, dois interpretadores que se destacam são $\text{X}\text{\TeX}$ e $\text{Lua}\text{\TeX}$. Além de serem mais rápidos, trazem implementações notáveis como seleção de fontes do próprio sistema (o $\text{pdf}\text{\TeX}$ não faz isso) ou interação com a linguagem de programação (brasileira) Lua (veja a Figura 4). A saída de cada um deles é o PDF, diretamente.

Em nossos sistemas, ao usarmos $\text{X}\text{\TeX}$ com \TeX usamos o atalho $\text{X}\text{\TeX}$. Da mesma forma, $\text{Lua}\text{\TeX}$ com \TeX é simplificado por $\text{Lua}\text{\TeX}$.

Obviamente, nesse minicurso, estamos interessados apenas no formato \TeX . Também, por mera preferência, usaremos como interpretador o $\text{Lua}\text{\TeX}$. Portanto, "compilaremos" nossos arquivos com `lualatex`.

Obtamos por usar `lualatex`, pois é o sucessor natural do `pdflatex`.⁴

**Figura 3.** Poderosa linguagem de programação (clique na imagem para saber mais)

³ falaremos sobre editores para \TeX mais abaixo.

Saiba mais

De acordo com Joseph Wright, usamos a palavra "compilação" herdada da computação, mas, uma palavra mais adequada seria "composição tipográfica" (`learnlatex`).

⁴ Confira essa informação no site <http://www.luatex.org/>

**Mudanças à vista ...**

Isso pode mudar algumas coisas para quem é acostumado a usar `pdflatex`, mas explicaremos as diferenças ao longo desse minicurso.

Geralmente chamamos esses "atalhos" de **compiladores**⁵.

Veja a Tabela 1 para comparação entre *compiladores*, *engine* e *format*.

⁵ Para mais detalhes, veja `lualatex-doc`.

Compilador	Interpretador	Formato
<code>tex</code>	<code>TeX</code>	plain <code>TeX</code>
<code>pdftex</code>	<code>pdfTeX</code>	plain <code>TeX</code>
<code>xetex</code>	<code>XeTeX</code>	plain <code>TeX</code>
<code>latex</code>	<code>pdfTeX</code>	<code>LATeX</code>
<code>pdflatex</code>	<code>pdfTeX</code>	<code>LATeX</code>
<code>xelatex</code>	<code>XeTeX</code>	<code>LATeX</code>
<code>lualatex</code>	<code>LuaTeX</code>	<code>LATeX</code>
<code>texexec</code>	<code>pdfTeX</code>	<code>ConTeXt</code>
<code>texexec --xtx</code>	<code>XeTeX</code>	<code>ConTeXt</code>
<code>context</code>	<code>LuaTeX</code>	<code>ConTeXt</code>

Tabela 1. Dando "nome aos bois"

Como esse texto é uma tentativa de introdução ao `LATeX`, não chegaremos nem perto do que gostaríamos de expor sobre essa simbiose entre uma linguagem de *programação* (Lua) integrada harmoniosamente com uma linguagem de *marcação* (`LATeX`). Basicamente, o código do `TeX` foi reescrito em Lua (uma linguagem de programação com muita relevância internacional, desenvolvida por brasileiros, na PUC-RJ), formando assim o `LuaTeX`. Houve, intencionalmente, a projeção para que esse *interpretador* fosse compatível com versões anteriores do `pdfTeX`, tornando o `LuaTeX` seu substituto natural, visto que é mais rápido e moderno.

Para encerrarmos essa subseção, é importante destacar a estabilidade do `LATeX`. Basicamente temos duas versões: uma antes de 1993, a saber `LATeX 2.09`; e, a de 1994 até HOJE, a saber `LATeX 2 ϵ` . Isso é interessante, pois os códigos são praticamente preservados ao longo do tempo: você poderia rodar um código em `LATeX` de 20 anos atrás sem muitos problemas.

Todavia, quando falamos em `LATeX`, hoje, estamos nos referindo às funcionalidades trazidas na versão `LATeX 2 ϵ` .

As atualizações são constantes, mas sem muitas mudanças estruturais. Todavia, está em desenvolvimento uma terceira versão do `LATeX`, a saber, `LATeX3` (veja a Figura 5).

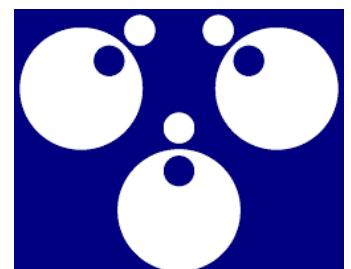


Figura 4. Saiba mais sobre o `LuaTeX` acessando o link: <http://www.luatex.org>



Figura 5. Saiba mais sobre esse fantástico projeto no link: <https://www.latex-project.org>

1.4 Como colocar o L^AT_EX em meu computador: as Distribuições

Para instalar localmente (em nosso computador) os *interpretadores*, precisamos das **distribuições**.

As principais são:

Distribuições	Sistema	Download/Instalação
MiK ^T E _X	Windows ou GNU/Linux ou Mac OS	https://miktex.org/download
T _E X Live	GNU/Linux ou Windows	https://www.tug.org/texlive/acquire-netinstall.html
MacT _E X	Mac OS	https://www.tug.org/mactex/

A distribuição MacT_EX contém todo o T_EX Live e adições específicas para Mac OS.

O T_EX Live completo precisa de, aproximadamente, 5GB de espaço em disco. Ele instala TODOS os pacotes disponíveis para L^AT_EX. Como hoje em dia a capacidade de disco é relativamente grande, a instalação completa pode ser interessante, caso você não queira depender mais da internet para instalação de futuros pacotes.

Mas, se você quiser usar apenas o necessário e manter uma distribuição mínima em seu computador, instalando futuros pacotes à medida que for precisando deles, talvez a distribuição MiK^TE_X seja a mais adequada.



Atenção!

O processo de instalação depende do sistema operacional e não será tratado nesse texto, visto que usaremos distribuições ONLINE dos interpretadores de T_EX.

Saiba mais

Para GNU/Linux existem outras opções de instalação que economizam espaço. Tudo dependerá da necessidade de cada um:
[texlive-latex-base](#)
[texlive-latex-recommended](#)
[texlive-pictures](#)
[texlive-fonts-recommended](#)
[texlive](#)
[texlive-plain-generic](#)
[texlive-latex-extra](#)
 Para mais informações veja essa discussão no [TeX SE](#)

2 Sim ...mas como eu uso?

Certo ... já temos uma distribuição T_EX, e agora? Basicamente, precisamos:

- *escrever*, usando o formato L^AT_EX, o que desejamos num arquivo de texto com extensão `.tex`;
- *compilar*, ou seja, compor tipograficamente o arquivo `.tex`, produzindo um arquivo `.pdf`; e, nesse ponto, usaremos o compilador `lualatex`;
- *visualizar* o arquivo `.pdf`; nesse ponto, o visualizador de PDF é de escolha pessoal.

É uma boa prática salvar os arquivos `.tex` com nomes **sem acentuação** ou **caracteres especiais** do teclado (%,\$,* etc.). E, se o nome do arquivo for

composto por mais de uma palavra, é aconselhável também **não deixar espaço entre elas**.

Por exemplo, suponha que você esteja escrevendo um artigo sobre *números complexos*. Seu arquivo principal não deve ser nomeado assim:

Artigo! Números Complexos.tex,

Você deve retirar o *ponto de exclamação* e o *acento agudo*, bem como retirar os espaços entre as palavras. Nesse último ponto, pode-se usar o *camelCase*, ou *snake_case* ou separar as palavras-com-traço. São aceitáveis qualquer das seguintes possibilidades (note que há uma preferência por letras minúsculas):

```
artigoNumerosComplexos.tex
artigo-numeros-complexos.tex
artigo_numeros_complexos.tex
artigo_numeros-complexos.tex
```

Particularmente, prefiro essa última opção: onde se mistura o *snake_case* com os traços. Tento manter algum padrão antes do *underline* e o nome do documento que estou escrevendo, separo por traços, caso seja necessário. Essa abordagem pode ser interessante se você estiver trabalhando com um arquivo principal (onde geralmente ocorre a compilação final) e muitos outros arquivos que serão incluídos no principal.

É uma forma de escrever ... Há quem prefira escrever tudo em um único arquivo .tex!

Falaremos sobre manipulação de vários arquivos mais à frente, porém, apenas para exemplificar a ideia, suponha que esse meu artigo (*artigo_numeros-complexos.tex*) sobre números complexos seja composto de cinco seções: introdução histórica; formas de representar um número complexo; raízes de um número complexo; o Teorema Fundamental da Álgebra; e, conclusões. Então, desejamos escrever essas seções em arquivos separados e "incluir-los" no arquivo principal, paulatinamente, por meio de compilações SOMENTE no arquivo principal.



Fica a dica!

Muitos nomeiam esses "arquivos principais", ou seja, onde ocorre a compilação, de *main.tex* (ou *master.tex*), que significa, em inglês, "principal", "importante", etc. ("senhor", "dominador", etc., para *master*).

Para esse minicurso, à jato, começaremos com um arquivo principal, nomeado por "*main.tex*", no qual escreveremos o conteúdo desejado inserindo as partes que o compõem, aos poucos.

Saiba mais

Uma forma de nomear esses arquivos seria, respectivamente:
01_intro-historica.tex;
02Representacoes.tex;
03_raizes.tex;
04_teo-fundamental.tex;
05_conclusao.tex.

Notem que dei a numeração no início, separando-a com *underline* e nomeei os arquivos de maneira que lembre-me a seção onde me encontro.

2.1 Como fazemos a "compilação"?

Bom ... ainda aprenderemos como escrever em L^AT_EX, não se preocupem, mas, antes, vamos aprender como compomos tipograficamente o texto.

Já vimos que precisamos escrever o conteúdo do texto em um arquivo de extensão .tex e compilarmos com lualatex ... Mas, como fazemos isso?

Ora, a escrita do texto pode ser feita em qualquer **editor de texto** de sua preferência. Pode-se usar um "bloco de notas"(para usuários de Windows), ou o "evince"(para usuários de GNU/Linux), por exemplo. Apenas deve-se lembrar em salvar o arquivo com extensão .tex.

Também é aconselhável manter o arquivo main.tex em algum diretório (pasta) nomeado adequadamente. Isso se deve ao fato de que, ao compilarmos, arquivos secundários são gerados (.aux, .log, etc.), o que pode gerar certa "bagunça".

Então, suponha que você tenha escrito um belo texto, cheio de equações matemáticas, num arquivo denominado main.tex, salvo em um diretório por nome "artigo_tcc/". Feito isso, abra o **terminal** (no Windows seria o "Prompt de Comando") dentro do diretório "artigo_tcc/" e digite o seguinte comando:



Usando o terminal para compilar

```
1 lualatex main.tex
```



Figura 6. Arquivos gerados numa compilação simples

Pelo menos três arquivos serão produzidos: um arquivo em [.pdf], que é a saída desejada; um arquivo .aux, essencial em referências cruzadas, por exemplo; e, um arquivo .log, o qual é um registro detalhado de tudo o que ocorreu na compilação, inclusive possíveis erros. Pode aparecer mais arquivos durante a compilação! Tudo dependerá de quão complexo é seu texto (se possui sumário; index; lista de tabela; lista de figura; glossário; etc). A Tabela 2 nos mostra alguns desses arquivos:

Tabela 2. Alguns dos possíveis arquivos gerados durante a compilação

Extensão	Descrição
.log	registro detalhado sobre a compilação, inclusive erros
.aux	registra processos intermediários, como referências cruzadas
.toc	serve para produção do índice
.lof	serve para produção da lista de figuras
.lot	serve para produção da lista de tabelas

Não tenha medo do terminal! Ele é seu amigo! Trabalhar usando "linha de comando", pelo terminal, é uma maneira de você comunicar-se com o computador de maneira rápida, direta e livre de distrações. O comando lualatex

`main.tex` simplesmente diz: "Olha ... componha tipograficamente meu texto que está no arquivo `main.tex`, usando o interpretador `Luat\TeX`, no formato `\TeX`".

Mas, se você quiser mais funcionalidades na digitação (autocomplete, identação automática, etc.) que um "bloco de texto" não oferece; bem como compilar sem usar o terminal, você precisará de um **editor de texto** especializado ou uma IDE (*Integrated Development Environment*).

2.2 Editores para \LaTeX

2.2.1 O \TeX é outro Word?

Um detalhe que precisamos ter em mente: a natureza dos editores para \TeX é geralmente diferente dos editores de texto como *Microsoft Word* ou *LibreOffice Writer*. Esses últimos editores são denominados WYSIWYG (acrônimo para a frase *What You See Is What You Get*). Em uma tradução literal livre significa "O que você vê é o que você tem", ou seja, o que você vê na tela de seu editor, é o que aparecerá na impressão. No \TeX não é bem assim ... Você vê comandos misturados com texto normal, porém o resultado sairá diferente daquilo que estará vendo em seu editor! Ficará mais bonito o resultado, não se preocupe. =)

Poderíamos listar argumentos sobre a superioridade do \TeX , comparando-o ao *Word* (por exemplo), mas isso não será feito. Cada um tem suas peculiaridades: vantajosas ou não. Tudo dependerá da finalidade do uso! Particularmente, tenho encontrado no \TeX um ambiente próprio para escrita matemática de forma rápida, bonita e gratuita! Textos longos, cheios de capítulos, ou documentos personalizados — como esse texto, exigiria um esforço e tempo tais, que nem passa por minha mente escrevê-los no *Word* (você já tentou criar um simples sumário no *Word*? Ou uma simples equação como esta: $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$, em menos de 10 segundos?)

Com isso em mente, é importante a aquisição de uma linguagem básica, mas, antes disso, vamos falar um pouco sobre os editores para \TeX .

2.2.2 Indicações de editores para \TeX

As possibilidades de editores são realmente vastas. Você pode olhar uma comparação entre eles nesse link:

https://en.wikipedia.org/wiki/Comparison_of_TeX_editors

Podemos usar editores **dedicados** ao \TeX ou editores mais **genéricos**.

No que se refere aos editores dedicados, listamos:

Saiba mais

"WYSIWYG ... O termo é usado para classificar ferramentas de edição e desenvolvimento que permitem visualizar, em tempo real, exatamente aquilo que será publicado ou impresso" (TECMUNDO, acesso em 03/02/2022).

Se você quiser saber como pronunciar corretamente essa sigla, acesse: [How to pronounce WYSIWYG](#).

- **TeXstudio**; se destaca por ser multiplataforma e com muitos recursos.

- **TeXnicCenter**; é voltado para usuários de Windows. Integra-se muito naturalmente ao sistema (e ainda indica o **Sumatra** para visualização do PDF).
- **LyX**; esse é um editor diferente dos citados anteriormente; pois renderiza as equações e estruturas do texto de forma quase imediata. É o mais próximo, para \LaTeX , de editores WYSIWYG.

Agora, no que se refere aos editores mais genéricos, ou seja, não dedicados ao \LaTeX , mas que com o uso de *plugins* essa realidade muda, destacam-se:

- **GNU Emacs**, junto com **AUC \TeX** ;
- **Atom**, junto com **atom-latex**;
- **Neovim**, com **VimTeX**;
- **VSCode** com **\TeX Workshop**;

Obviamente, para o iniciante, é indicado um editor dedicado ao \LaTeX .⁶

O *plugin* do \TeX Workshop faz o processo de compilação de uma forma um pouco diferente: ele usa uma ferramenta chamada **latexmk** que nada mais é que um *script* que automatiza muitos processos. Por exemplo, quando o documento é complexo; com referências cruzadas; sumário; glossário; lista de tabelas; lista de figuras; etc. é necessário mais de um processo de compilação. Esse *plugin* faz isso tudo parecer mágica. Ele vem, por padrão, configurado para **pdflatex**, mas pode ser modificado o interpretador de maneira bem simples.

Uma ferramenta parecida com o **latexmk**, mas que precisa de diretivas (você precisa "dizer" para ele os passos que deseja executar) é o **arara**. Ele foi desenvolvido por um brasileiro, Paulo Cereda e possui ferramentas de limpeza de arquivos auxiliares; rapidez na segunda compilação; produção de arquivo *draft* (um rascunho que não gera o pdf, o que torna as coisas muito mais rápidas); etc. Falaremos sobre ele no Apêndice, pois vale muito a pena conhecer essa ferramenta!

⁶ Particularmente, uso o VS-Code com o *plugin* do \TeX Workshop.

