

O (PEQUENO) MANUAL DE L^AT_EX DO LINGUISTA CONTEMPORÂNEO

versão beta

Rafael Luis Beraldo

31 de agosto de 2010

Sumário

1	Introdução	2
2	A Filosofia do \LaTeX	4
2.1	Filosofia do código	4
2.2	Filosofia das possibilidades	6
2.3	Filosofia <i>open source</i>	6
3	Tipografia	8
3.1	Aspas	8
3.2	Reticências	9
3.3	Hífens, meia-risca, travessões e menos	9
3.3.1	...e ponto final!	10
3.4	Ele faz, mas você não vê	10
4	Recursos avançados para linguistas	11
4.1	TIPA	11
4.2	Árvores	12
4.3	<i>Attribute-value matrices</i>	13
5	Terminando	15
6	Links	16

Capítulo 1

Introdução

O \LaTeX (pronunciado $/\text{'lertek}/$) é uma **linguagem de marcação** muito usada por matemáticos, físicos e cientistas da computação. Para eles, o principal atrativo é a facilidade da composição de fórmulas matemáticas; em verdade, o \LaTeX é o padrão internacional¹ de equações matemáticas em *plain text*. Enquanto os programas WYSIWYG (acrônimo de What You See Is What You Get) focam em como o documento ficará, à semelhança das máquinas de escrever, o \LaTeX te dá a possibilidade de organizar seu documento logicamente em seções, dividi-lo em vários arquivos, e usar ferramentas avançadas.

O foco deste manual são justamente essas ferramentas. Na primeira parte, apresento uma série de razões (“filosóficas”) para o uso de \LaTeX pelos linguistas; depois falo de alguns tópicos de **tipografia**, tradição que foi praticamente soterrada com a popularização do *Microsoft Word* e semelhantes; finalmente, me foco nos recursos para linguistas.

Acredito que essa organização seja necessária por dois motivos:

- Dentro da Linguística encontrei muito pouco uso de \LaTeX . As únicas iniciativas que vi são internacionais. Acredito que uso de uma ferramenta científica em detrimento daqueles que são, na prática, amadoras, é completamente justificável e traria melhorias;
- Há muito pouca gente nas Humanas que conheça a filosofia do uso dos computadores, o que implica em um uso e visão insatisfatórios. A maioria se vale de soluções-padrão, que não seguem a “tradição de uso” do computador; a tentativa de “tornar fácil” uma ferramenta que é, *per se*, difícil, causa muitos problemas que não estão sendo discutidos. Provavelmente isso não seja culpa dos usuários: o único caminho que conhecem é este. Introduzo uma filosofia do uso de \LaTeX , que se liga muito fortemente à filosofia do software livre em geral, como uma *introdução a um novo modelo de uso e de pensamento*;

¹Isto é, \LaTeX é uma convenção na tipografia de fórmulas matemáticas.

Haveria um terceiro motivo se fôssemos levar em consideração as inconsistências causadas pelo uso das normas da ABNT, mas prefiro citá-los vagamente e ocasionalmente a desviar o foco deste e causar indisposições com os defensores e usuários da “norma”.

Para aprender \LaTeX , sugiro que você use o sistema operacional GNU/Linux² e leia este documento: *The not too short introduction to \LaTeX 2 ϵ* (<http://tobi.oetiker.ch/lshort/lshort.pdf>) ou a versão em português, aqui: *Uma não tão pequena introdução ao \LaTeX 2 ϵ* (<http://www.tug.org/texlive/Contents/live/texmf-doc/doc/portuguese/lshort-portuguese/ptlshort.pdf>).

Qualquer explicação extra que você precise pode ser encontrado na Internet. A maioria das dúvidas que você terá já foi respondida, especialmente se são dúvidas elementares, de iniciante. Por isso eu recomendo fortemente que aquele que deseje usar o computador como ferramenta procure conhecer as listas de e-mail, os repositórios de documentação, fóruns, canais de IRC. Isso será útil durante seu aprendizado de \LaTeX e qualquer outro programa de código aberto.

Agora podemos começar.

²Por um motivo simples: instalá-lo no Linux é extremamente trivial.

Capítulo 2

A Filosofia do L^AT_EX

2.1 Filosofia do código

Eu não vou falar sobre a história do L^AT_EX. Isso você encontra naquele texto, *Not too short*, que eu recomendei. Vou falar sobre o que eu aprendi *na prática*, discutindo com usuários de L^AT_EX e usando.

Os documentos gerados usando essa linguagem tendem a ser muito mais organizados, lógicos, bem formatados e leves. O texto no L^AT_EX é **sempre justificado**. Ao contrário de editores como o *Microsoft Word*, que são visuais, o L^AT_EX se foca no *conteúdo*, sem a distração visual¹.

Começemos pela parte mais prática da filosofia dessa linguagem de marcação de texto: seu uso. Ela funciona de forma lógica. Um documento geralmente se organiza assim:

```
\begin{document}
```

```
\section{Primeira seção}
```

Essa é a primeira seção de seu texto. Você pode criar subseções facilmente, fazendo um

```
\subsection{Primeira subseção}
```

sem nem se preocupar com a numeração.

¹Isso pode parecer meio intangível, falso, mas conversei com um professor da Universidade de Amsterdam, via e-mail, e ele me disse que sua tese de mestrado tinha quase o dobro do tamanho das teses que as pessoas geralmente fazem por lá. O motivo é que um arquivo muito grande em editores visuais vai ficando confuso, o programa fica lento, a organização é precária (por causa da filosofia destes editores) etc.

A numeração das páginas também é automática. O texto é por padrão justificado, mas você pode facilmente centralizá-lo assim:

```
\begin{center}
E este será um texto centralizado.
\end{center}
```

```
\section{Segunda seção}
```

Como você pode perceber, tudo é muito lógico e `\emph{simples}`. A palavra ‘‘simples’’, na frase anterior, sairá enfatizada, ou seja, em itálico. `\textbf{Já todo esta frase será em negrito.}`

Fazer notas de rodapé é muito fácil. Basta usar o comando `\footnote{Conteúdo da nota.}` e pronto! Sem se preocupar com numerações nem nada.

```
\end{document}
```

Talvez agora dizer que o \LaTeX se foca no conteúdo faça mais sentido. Se você precisa adicionar mais uma seção, usamos o comando `\section`. Para o sumário, basta um `\tableofcontents`, que usa todas as seções e subseções ou capítulos e subcapítulos e cria um índice automaticamente. As numerações de seção, as formatações e quebras de página, etc., tudo é feito automaticamente.

A segunda coisa que o \LaTeX resolve é a famosa **incompatibilidade**. Depois que você escreve seu arquivo `.tex`, você deve passá-lo por um processo conhecido como **compilação**. Isso significa que um programa pegará o código e o transformará naquilo que você mandou. O arquivo final é geralmente um elegante PDF. Você pode abri-lo em qualquer computador que tenha um leitor de PDF — e praticamente todos têm.

Isso *acaba* com o velho problema de incompatibilidade, que é quando você faz o arquivo numa versão de um programa, e tenta abri-lo numa mais velha ou mais nova, e perde toda sua formatação, ou o programa recusa-se a lê-lo etc. O \LaTeX também pode gerar apresentações de slide, mas digo isso a título de curiosidade. Se você tem vontade de aprender, sugiro que leia sobre o **Beamer** ou o **Prosper**, dois pacotes para se criar apresentações.

Vamos, agora, subir um pouco no degrau da filosofia do \LaTeX .

2.2 Filosofia das possibilidades

Qual é o suporte que as ferramentas que você usa atualmente têm ao alfabeto fonético, árvores como as usadas nos esquemas do gerativismo, caracteres e símbolos exóticos, alta qualidade tipográfica, diagramas de vogais, bibliografias, dissertações, referências cruzadas, AVMs (*attribute-value matrices*)?

O \LaTeX suporta. Nele temos ferramentas muito boas que inexistem em outros editores. Isso centraliza a edição, torna mais fácil o controle dos arquivos que você cria, evita o uso de imagens (criar um diagrama de vogais à mão, por exemplo, e depois colocá-lo no documento em forma de imagem), que são e tornam o documento pesado. Cada uma dessas ferramentas têm uma sintaxe própria, e seguem o padrão da sintaxe do \LaTeX . Uma vez familiarizado com ela, o uso se torna natural, orgânico.

Possuir tantos recursos é um ponto muito importante sobre o \LaTeX ; a qualidade deles é atestada: são mais de vinte anos de vida. Vinte anos de melhoria.

2.3 Filosofia *open source*

Esse é o ponto mais interessante. Em primeiro lugar, o \LaTeX é *aberto*. As primeiras implicações que isso causa é que ele é grátis — não estou tentando vender nada para você. Mas há implicações mais importantes que o fato dele ser gratuito.

O modelo *open source*², ou seja, *código livre*, garante a distribuição de um programa e seu código. Quem se interessar em modificar o programa, entender como ele funciona, ou melhorá-lo, está livre e é encorajado a fazê-lo. Como o código do programa passa por muitas mãos, no caso do *open source*, frequentemente notam-se erros ou melhorias possíveis. Se você encontrou algum erro, existem formulários intuitivos para se preencher e reportar o erro, que será avaliado por toda a comunidade. Este modelo de escrita de programas tem se provado o mais eficiente já concebido. Por isso mesmo o \LaTeX tem tanto suporte a coisas interessantes para a Linguística (entre outras áreas), enquanto outros softwares não. Além disso, o \LaTeX é muito mais estável que qualquer outro processador de texto que você conhece, e evolui rapidamente.

Quando você escreve um texto no *Word* ou *OpenOffice.org Writer*, por exemplo, o que acontece é que cada formatação que você faz é, na verdade, um código. A diferença entre eles e o \LaTeX é que escodem esse código e o processam no momento em que você está vendo. Por isso documentos muito grandes são simplesmente inviáveis: o programa tem de processar muita coisa “ao vivo”, e é muito fácil ocorrerem problemas devidos a tanto processamento.

²Em inglês, softwares desenvolvidos seguindo este modelo são chamados de *free software*. Quando traduzimos para o português, fica “software livre”, porque o *free* não é de “grátis”; como diz uma ladainha famosa, *free as in freedom*.

O \LaTeX possui versões para Windows, para Linux, para Mac. Você pode editar seu texto onde estiver, compilá-lo onde estiver, e visualizá-lo onde estiver. Você não é obrigado a usar este ou aquele sistema operacional, nem este ou aquele programa. Existem aplicativos gratuitos muito bons para se trabalhar com \LaTeX , assim como existem os pagos.

Algo muito suspeito que costumam fazer as empresas de software são programas pesados, que exigem muito do computador (e, é claro, do bolso do usuário, que tem de fazer upgrades que poderiam ser tornados desnecessários por uma programação mais inteligente e menos exigente). Por outro lado, o software livre costuma ser leve e otimizado. Naturalmente alguns aplicativos são pesados por natureza, mas não há sentido em precisar de um computador potente para atividades triviais como editar um texto ou navegar na Internet.

Capítulo 3

Tipografia (ou Porque Não Podemos Esquecer Seiscentos Anos de Tradição)

Agora você sabe usar o \LaTeX , está quase pronto para aprender os recursos avançados para linguistas. Porém, eu quero parar teus saltos de progresso enormes para falar sobre... Detalhes. Existem alguns padrões que remontam à tradição da tipografia, mas falta informação sobre isso. O que farei aqui é comentar um pouco sobre questões que devem ser respeitadas quando editamos um texto.

Essa é provavelmente uma seção que vai crescer muito neste documento, conforme eu for aprendendo mais. Tenho apenas um ano de experiência com a ferramenta.

3.1 Aspas

Nas máquinas de escrever, as aspas eram assim: ". Isso servia para *economizar espaço*. Com uma única aspa, neutra, você dava conta do recado.

Porém... Isso é errado. Nos computadores existem combinação que dão as aspas corretas.

Modelo do Teclado ¹	<i>Input</i>	<i>Output</i>
Internacional	<code>altgr+(</code>	‘
	<code>altgr+)</code>	’
	<code>altgr+shift+{</code>	“
	<code>altgr+shift+}</code>	”
ABNT ₂	<code>altgr+shift+v</code>	‘
	<code>altgr+shift+b</code>	’
	<code>altgr+v</code>	“
	<code>altgr+b</code>	”

Usando as aspas erradas, nós teríamos “estes caracteres” ao invés “destes caracteres”, que são muito mais bonitos. Conseguimos as aspas no \LaTeX dessa maneira:

‘‘Ah, agora sim! Tudo certo!’’

Os caracteres são, respectivamente, dois acentos graves e dois agudos.

3.2 Reticências

As reticências são feitas usando o comando `\ldots`. Isso acontece por um motivo muito simples: as reticências são, na verdade, um único caractere, com espaçamento diferente entre cada ponto do que conseguimos usando os clássicos “três pontinhos”. Veja:

<i>Input</i>	<i>Output</i>
<code>...</code>	...
<code>\ldots</code>	...

3.3 Hífens, meia-risca, travessões e menos

Aqui uma tabela para referência rápida: como usar hífens, meia-riscas, travessões e sinais de menos².

Tipo	<i>Input</i>	<i>Output</i>	Exemplo
hífen	<code>-</code>	-	Eis meu guarda-chuva!
meia-risca	<code>--</code>	–	Leia as páginas 5–23.
travessão	<code>---</code>	—	Eu disse que era assim — e não foi por mal!
sinal de menos	<code>\$-\$</code>	−	Quanto dá 5 − 3?

Isso deveria ser feito fora do \LaTeX também. Para maior informações, leia esta página na Wikipédia.

²Esta tabela foi adaptada a partir do original encontrado em *A gentle introduction to \TeX* .

3.3.1 ... e ponto final!

O \LaTeX adiciona um espaço maior no fim de frases do que entre palavras. Isso deixa o texto mais legível. Mas existem duas ocasiões em que você deve ficar atento ao espaçamento. Veja os exemplos abaixo:

O Sr.~João lhe mandou esta
carta, Sra.~Fnord.

O Sr. João lhe mandou esta carta,
Sra. Fnord.

Isso é a ABNT. É, poderia ser melhor.

Isso é a ABNT\@. É, poderia
ser melhor.

A presença de um til ‘~’ faz com que não haja um espaçamento maior no fim das linhas. Já o \@ antes do ponto final diz ao \LaTeX que este é um fim de frase. Ele não faz este espaço quando o ponto final vem logo depois de uma letra maiúscula, como em siglas. Portanto, colocamos o \@ para que o espaçamento padrão de fim de frase seja respeitado.

Para ter espaços sempre iguais, use o comando `\frenchspacing`.

3.4 Ele faz, mas você não vê

Existe uma quantidade muito grande de detalhes que o \LaTeX leva em consideração, nos livrando de um trabalho que seria, de outra maneira, enorme. A justificação é automática e feita usando **hifenização**, coisa que o *Word* é incapaz de fazer.

Eu poderia falar bastante sobre esse assunto, mas deixo dois links de pessoas que entendem sobre tipografia. Leiam para vislumbrar o grau de refinamento que tem o \LaTeX .

- The Beauty of \LaTeX (<http://nitens.org/taraborelli/latex>), onde Dario Taraborelli compara a qualidade da formatação do \LaTeX em relação ao *Word*. Também dá exemplos do poder do processador, e tem uma grande quantidade de links relacionados;
- \LaTeX vs. Word vs. Writer (<http://oestrem.com/thingstwicetwice/2007/05/latex-vs-word-vs-writer/>), por Eyolf Østrem. Neste texto ele compara a qualidade entre o \LaTeX , o *Microsoft Word* e o *OpenOffice.org Writer*.

Você vai perceber que muitas “editoras” parecem nem sequer conhecer essas questões, o que é uma pena.

Capítulo 4

Recursos avançados para linguistas

Finalmente, chegamos ao ponto alto deste manual. Se não é o ponto alto, ao menos é o foco. Doravante vamos conhecer o poder do \LaTeX aplicado na Linguística.

4.1 TIPA

TIPA (Tokyo IPA) é um pacote criado por Fukui Rey para processar os caracteres do *International Phonetic Alphabet*, que é usado para transcrições fonéticas. Suporta até mesmo os caracteres considerados obsoletos pelas revisões de 1989 e 1996.

Para usá-lo, devemos adicionar a linha `\usepackage{tipa}` o preâmbulo do documento. Para usar o alfabeto fonético, escrevemos o comando `\textipa{ }`. Por exemplo:

Input: `\textipa{/kaza/}`

Output: `/kaza/`

Ou algo mais sofisticado, como:

Input: `\textipa{["Eksplo"neIS@n]}`

Output: `[.ɛksplə'neɪʃən]`

O pacote TIPA resolve muitos problemas. O primeiro deles é que é muito lento escrever símbolos fonéticos no *Word*, por exemplo: você tem que selecionar os símbolos, um por um, dentro de uma grande tabela.

Já com o TIPA, escrevemos seguindo uma convenção. O *output* segue esta convenção, que é descrita numa tabela muito parecida com a tabela clássica do IPA, que você pode ver neste link (<http://www.ling.ohio-state.edu/events/lcc/tutorials/tipachart/tipachart.pdf>).

Uma fonte *sans-serif* vem junto com o TIPA, acessíveis via

```
\textsf{\textipa{ }}
```

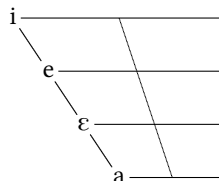
Podemos usar, ainda, o ambiente

```
\begin{IPA}
:
\end{IPA}
```

para escrever longos trechos de transcrição fonética.

O pacote TIPA inclui o pacote `vowel`, que permite desenhar a **tabela de vogais**:

```
\begin{vowel}
\putcvowel{i}{1}
\putcvowel{e}{2}
\putcvowel{\textipa{E}}{3}
\putcvowel{a}{4}
\end{vowel}
```



(A tabela de vogais acima está incompleta.) Para usar o ambiente `vowel`, você deve incluir, no preâmbulo, a linha `\usepackage{vowel}`.

O manual do pacote TIPA está no seguinte endereço: <http://www.l.u-tokyo.ac.jp/~f-kr/tipa/tipaman.pdf>

Slides de um curso muito completo disponíveis no site do *Center for Advanced Study in Theoretical Linguistics*, da *Universitetet i Tromsø* podem ser vistos no link <http://uit.no/getfile.php?PagelId=874&FileId=299>

4.2 Árvores

As árvores são usadas na Linguística, Lógica, Matemática, Ciências da Computação e outros campos. Existem softwares específicos para se criar árvores, mas eu não sei como eles funcionam. Achei muito mais simples, porém, usar o pacote `qtree`. Para usá-lo, você deve digitar, no preâmbulo, `\usepackage{qtree}`. Vejamos como funciona uma árvore em \LaTeX :