
L^AT_EX

para

[l[~]η. 'g^wis.tes]

Marcelo Ferreira
ferreira10@usp.br

Marcos Lopes
marcoslopes@usp.br

Francisco Merçon
franciscomercon@usp.br

DL-USP
Outubro de 2009



Esta obra está licenciada sob uma Licença Creative Commons.

Sumário

Prefácio	vi
1 Instalação	1
1.1 Instalação no Windows	1
1.2 Instalação no Linux	2
1.3 Alguns links úteis	3
2 Criando e Estruturando um Documento	5
2.1 Estrutura Básica	5
2.2 Processando um Documento	6
2.3 Digitando texto	7
2.3.1 Comentários	11
2.3.2 Caracteres especiais	11
2.4 Português	12
2.5 Fontes	12
2.6 Notas de Rodapé	13
2.7 Título, Autor e Data	14
2.8 Numeração de página	15
2.9 Espaçamento entre linhas	16
2.10 Divisão do texto	16
2.10.1 Seções, subseções,	16
2.10.2 Abstracts	17
2.10.3 Apêndice	17
2.11 Gerando um Sumário	17
3 Ambientes	19
3.1 Listas	19
3.2 Tabelas	21
3.3 Figuras	24
3.3.1 Alterando o tamanho da figura	25
3.3.2 Usando o ambiente “figure”	26
3.3.3 Inserindo legendas nas figuras	27
3.3.4 Inserindo mais de uma figura	30

3.3.5	Posição absoluta	31
3.4	Centralizar	32
3.5	Citações	33
3.6	Epígrafes	33
3.7	Versos	34
3.8	Cabeçalhos e rodapés	36
3.9	Cores	39
3.9.1	Cores básicas	39
3.9.2	Alterando cores	39
4	Aplicações Linguísticas	43
4.1	Símbolos Fonéticos	43
4.1.1	Instalação	43
4.1.2	Transcrevendo	44
4.1.3	Atalhos no teclado	45
4.1.4	Tabela completa dos símbolos IPA	45
4.1.5	Diacríticos	46
4.2	Símbolos Lógico-Semânticos	47
4.3	Símbolos utilizados em Semiótica	48
4.4	Outros símbolos	50
4.5	Árvores Sintáticas	50
4.6	Quadrados semióticos e Esquemas tensivos	52
4.7	Exemplos Numerados	57
4.7.1	Sub-Exemplos	57
4.7.2	Glosas	58
4.7.3	Referências Cruzadas	59
5	Apresentações Acadêmicas	61
5.1	Slides com <i>Beamer</i>	61
5.1.1	Recursos Básicos	61
5.1.2	Estruturação	65
5.1.3	Aparência	65
5.1.4	Handouts	66
5.2	Posters com <i>posterusp</i>	66
6	Referências Bibliográficas	71
6.1	Preliminares	71
6.2	Mãos às obras	72
6.2.1	Por que quatro compilações?	73
6.3	Bancos de Dados bibliográficos	74
6.3.1	Organização do arquivo bibliográfico	74
6.4	Como citar as obras no texto	75
6.4.1	Citações nos rodapés	76
6.5	Vantagens do uso do BIB _T E _X	76

6.6	Opcional: configuração do TeXnicCenter	77
6.7	Outros links úteis	77
7	Seu Texto em L^AT_EX	79
7.1	O que escrever no preâmbulo dos documentos?	79
7.1.1	Artigos	81
7.1.2	Documentos “oficiais” (pareceres, relatórios, etc.)	83
7.1.3	Cartas oficiais	85
7.1.4	Slides	87
7.1.5	Posters	89
7.1.6	Teses e livros	91
7.2	Erros na compilação	96
7.2.1	Tipos de erros	96
7.2.2	Erros comuns	96
7.2.3	Como rastrear e eliminar os erros	97
7.3	Algumas questões frequentes	98
7.3.1	Generalidades	98
7.3.2	Configuração da página	98
7.3.3	Interação do L ^A T _E X com outros programas	100
7.3.4	Outros recursos de uso acadêmico	102
7.4	Links (muito) úteis	104
7.5	Livros	104

Prefácio

Esta apostila é um breve curso sobre o uso do software LaTeX, que é um sistema profissional de tipografia, especialmente concebido para publicações científicas (artigos, livros, teses, etc.). Não é nossa intenção aqui fazer publicidade do sistema, mas cabe dizer que diversas das principais editoras acadêmicas mundo afora já publicam em LaTeX, bem como um número crescente entre as principais revistas internacionais na área de Linguística. Nossa principal motivação em elaborar este material, entretanto, foi a percepção de que professores e alunos muitas vezes têm dificuldades com a diagramação de textos e a inclusão de objetos gráficos como fontes fonéticas (ou em outros alfabetos), árvores sintáticas, diagramas, esquemas tensivos, glosas, etc., além de bibliografias, sumários, referências cruzadas e outros. Fora esses tópicos ligados à preparação de textos em papel, também trataremos da geração de slides no LaTeX e posters para congressos. O que está nas páginas seguintes é, portanto, um curso de LaTeX para linguistas.

Pretendemos revisar continuamente este material. Críticas, comentários e sugestões são mais que bem vindos!

Capítulo 1

Instalação

1.1 Instalação no Windows

1. Instale a implementação do \LaTeX para Windows, denominada MiKTeX, baixando e instalando o programa deste [link](#).

Na página que vai se abrir, clique o botão **Download** que aparece associado ao tópico “Basic MiKTeX 2.7 Installer”. Grave o arquivo em seu disco rígido e execute-o a seguir.

Durante a instalação, é recomendável manter todas as opções-padrão propostas.

2. Uma vez instalado o programa, será preciso declarar as línguas que você vai usar na criação de seus textos, por causa da hifenização. Para tanto, clique o botão *Iniciar* do Windows, e depois *Programas/MiKTeX 2.7/Settings/Languages*. Apesar de você poder incluir todas as línguas que quiser, é aconselhável ater-se àquelas que de fato serão usadas.
3. A seguir, **depois de ter instalado o MiKTeX**, baixe e instale o editor, denominado TeXnicCenter. Para iniciar o download do programa, clique neste [link](#).
4. Execute o TeXnicCenter. Durante sua primeira execução, ele deve perguntar o caminho (“path”) para os executáveis do MiKTeX. Se você realizou a instalação padrão deste último (isto é, sem alterar as sugestões apresentadas por ele), os executáveis estarão no diretório `C:\Arquivos de Programas\MiKTeX 2.7\miktex\bin`.

1.2 Instalação no Linux

A instalação do \LaTeX nos sistemas Linux é mais simples e rápida do que no Windows. Tudo o que você tem a fazer é abrir um terminal¹ e digitar as três linhas a seguir:

```
sudo apt-get update
sudo apt-get install texlive
sudo apt-get install kile
```

Com isso, o sistema já estará pronto para uso. É importante, no entanto, notar alguns pequenos detalhes.

Primeiro, se você preferir, também é possível instalar os pacotes do TeXLive e do Kile através do Gerenciador de Pacotes de sua distribuição do Linux, como o Synaptic (no Ubuntu) ou o Yum (no Fedora ou no Yellow Dog). Seu gerenciador também pode ser usado para instalar pacotes adicionais para o \LaTeX . Eis aqui alguns fortemente recomendados para linguistas:

- `texlive-lang-portuguese`
- `texlive-humanities`
- `tipa`
- `texlive-latex-extra`
- `texlive-latex-recommended`

Outra possibilidade seria instalar de uma vez todos os pacotes, substituindo a segunda linha de comando acima por `sudo apt-get install texlive-full`. Entretanto, isso ocuparia muito mais espaço em seu disco rígido, desnecessariamente.

A terceira linha de comando listada acima instala o Kile, que é um bom editor. Há muitos outros editores, mas o Kile é bastante completo e vem sendo constantemente atualizado. Observe, entretanto, que ele foi concebido para funcionar com o ambiente gráfico KDE. Se você usa outro ambiente (como Gnome, que é aquele por default do Ubuntu), talvez queira instalar pacotes adicionais para ter acesso a certos recursos, como visualização em PDF, Ajuda e Correção Ortográfica. Quanto ao visualizador de PDFs, é possível configurá-lo através do menu *Settings / Configure Kile . . . / Tools / Preview*, em “Select a configuration:”.

Um último detalhe sobre a configuração do Kile: pode ser que seja interessante para você gravar arquivos que mais tarde serão lidos por programas rodando em Windows XP. A codificação por default dos sistemas Linux

¹No Ubuntu, o terminal pode ser encontrado no menu *Aplicativos / Acessórios*.

(assim como dos Windows mais modernos, Vista e 7) é UTF-8, mas a do Windows XP é a antiga ISO 8859-1. Para que não haja problemas com os caracteres acentuados, você deve gravar e abrir seus arquivos nessa codificação. É possível configurar o Kile para fazer isso automaticamente, através do menu *Settings / Configure Kile. . . / Open-Save / Encoding*, escolhendo a opção “Western European (ISO 8859-1)”.

1.3 Alguns links úteis

[Usando L^AT_EX](#) Curso completo oferecido pelo Moodle / USP, incluindo um vídeo com instruções de instalação.

[Pacotes](#) para o L^AT_EX Eles estendem os recursos originais do sistema e facilitam a sua utilização. Podem ser instalados diretamente através dos gerenciadores de pacotes no Linux (como o Synaptic) ou por meio do MiKTeX no Windows (*Iniciar/Programas/MiKTeX 2.7/Settings/Packages*).

[Um manual de L^AT_EX em português](#)

[Seção sobre L^AT_EX dentre os Wikibooks da Wikipedia \(em inglês\)](#)

[The beauty of LaTeX](#). Compara alguns dos recursos tipográficos do L^AT_EX com os processadores de texto “WYSIWYG”, como o MS-Word.

Capítulo 2

Criando e Estruturando um Documento

2.1 Estrutura Básica

Todo documento a ser processado se inicia por uma linha como a seguinte:

```
\documentclass{}
```

Entre as chaves deve-se declarar o tipo de documento em questão. A opção mais comum e que tomaremos por base em nossos primeiros exemplos é **article**.

```
\documentclass{article}
```

Nesta mesma linha, pode-se ainda especificar (por exemplo) o tamanho da fonte e da folha em que o texto final será gerado. Essas especificações devem vir antes das chaves, entre colchetes e separadas por vírgula. Por exemplo, para fonte tamanho 12 e papel do tipo a4, declara-se:

```
\documentclass[12pt,a4paper]{article}
```

Se nada for especificado, o latex usará as opções *default*, que são fonte tamanho 10pt e papel tipo carta.

Logo após essa primeira linha, inicia-se o que se chama de *preâmbulo*, que é o espaço destinado à inserção de alguns parâmetros gerais de formatação e aos chamados *pacotes*, que são programas destinados a tarefas específicas, como a inserção de exemplos numerados, o desenho de árvores sintáticas, rodapés de páginas sofisticados, etc ... Se nada disso for relevante para seu texto, esse espaço pode permanecer vazio.

Apenas para ilustrar, aqui está um exemplo do uso de um pacote a ser declarado no preâmbulo de um documento:

6 CAPÍTULO 2. CRIANDO E ESTRUTURANDO UM DOCUMENTO

```
\usepackage{fullpage}
```

O pacote `fullpage` impõe ao documento margens de aproximadamente 2,5 cm (o default do Latex são margens mais largas).

Após o preâmbulo, inicia-se o *corpo do documento*, que é onde você digitará o conteúdo de seu texto. O início do corpo do documento é indicado pela seguinte linha:

```
\begin{document}
```

O fim do corpo do documento é indicado pela seguinte linha:

```
\end{document}
```

Tudo o que você digitar depois dessa linha será ignorado durante o processamento do documento.

Um documento mínimo terá então a seguinte forma:

```
\documentclass{article}
\begin{document}

Oi!

\end{document}
```

Se fizéssemos uso das opções sugeridas mais acima teríamos

```
\documentclass[12pt,a4paper]{article}

\usepackage{fullpage}

\begin{document}

Oi!

\end{document}
```

2.2 Processando um Documento

Um documento a ser processado deve ser gravado como um arquivo com extensão `.tex`. Nos editores de texto voltados especialmente para o Latex – como o TeXnicenter, por exemplo – essa já é a extensão padrão em que os documentos são gravados.

Esse arquivo com a extensão `.tex` é a fonte a partir da qual será gerado o arquivo final com o conteúdo de seu documento formatado. Esse arquivo

resultante do processamento será gravado automaticamente no mesmo diretório em que se encontra o arquivo fonte. Ele será gravado com o mesmo nome deste arquivo, porém com uma extensão diferente. Essa extensão pode ser **.dvi**, **.ps** ou **.pdf**. Aqui, assumiremos que o output desejado **.pdf** e portanto falaremos apenas no tipo de processamento que gera tal arquivo diretamente do arquivo **.tex**. Assumiremos ainda que você está utilizando o **TeXnicCenter** como editor de texto.

Dito tudo isso, efetuar o processamento é muito simples: tendo digitado e gravado seu documento, selecione na barra de ferramentas do TeXnicCenter a opção *Latex -> pdf*. Em seguida, basta clicar o botão *Build and view current file* e o arquivo **.pdf** será gerado e logo em seguida aberto na tela de seu computador.

2.3 Digitando texto

A digitação do conteúdo de um texto (input) é feita da maneira usual via teclado, mas algumas ressalvas devem ser feitas a fim de se evitar surpresas no resultado final (output).

A primeira coisa a se notar é que mais de um espaço em branco entre palavras, ou espaços em branco no início de uma linha do input, não tem nenhum efeito no espaçamento entre palavras no output.

Eu sou a primeira sentença.	Eu sou a primeira sentença.
Eu sou a segunda sentença.	Eu sou a segunda sentença.
E eu sou a terceira.	E eu sou a terceira.

Se por algum motivo você quiser um maior espaçamento entre algumas palavras, utilize a barra invertida (`\`) seguida de um espaço em branco:

Eu sou a primeira sentença.	Eu sou a primeira sentença.
Eu \ sou \ \ a \ \ \ segunda.	Eu sou a segunda.

Um parágrafo novo é indicado por uma linha em branco no input.

Eu estou neste parágrafo.	Eu estou neste parágrafo.
E eu estou em outro.	E eu estou em outro.

O \LaTeX insere automaticamente um recuo no início de cada parágrafo (a exceção são os parágrafos que iniciam uma seção, como veremos mais adiante). Se você quiser que um parágrafo não inicie com esse recuo, utilize o comando `\noindent`:

8 CAPÍTULO 2. CRIANDO E ESTRUTURANDO UM DOCUMENTO

Eu sou o fim de um parágrafo.	Eu sou o fim de um parágrafo.
<code>\noindent</code>	E eu não tenho recuo.
E eu não tenho recuo.	

Note que é crucial soltar uma linha para mudar de parágrafo:

Eu estou neste parágrafo.	Eu estou neste parágrafo. Eu tam-
Eu também.	bém.

Por outro lado, deixar mais de uma linha em branco tem o mesmo efeito que deixar apenas uma.

Eu estou neste parágrafo.	Eu estou neste parágrafo.
	E eu estou em outro.
E eu estou em outro.	

Se você quiser apenas mudar de linha, utilize `\\`:

Eu estou nesta linha.\\ Eu não.	Eu estou nesta linha.
	Eu não.

E se você quiser uma linha em branco (ou várias linhas em branco) entre duas linhas de texto, utilize `\\` (ou `\\\\`, ...):

Eu estou aqui.\\ E eu aqui.	Eu estou aqui.
	E eu aqui.

A fim de determinar arbitrariamente um espaço em branco vertical no texto, utilize o comando `vspace{x}`, onde x é a medida que se quer “saltar”.

Eu estou aqui.	Eu estou aqui.
<code>\vspace{1cm}</code>	
E eu aqui.	E eu aqui.

Analogamente, para os espaços horizontais, existe o comando `\hspace`.

Eu estou aqui. <code>\hspace{3cm}</code>	Eu estou aqui.	E eu aqui.
E eu aqui.\\		

Você também pode preencher horizontal ou verticalmente uma página através dos comandos `\hfill` e `vfill`. Eles podem ser úteis quando se deseja criar um efeito de simetria da distribuição do texto na página, o que é bastante comum em páginas de rosto, por exemplo.

```
texto \hfill texto \\
texto texto \hfill texto texto \\
texto texto texto \hfill texto texto texto \\
```

```
texto                                     texto
texto texto                               texto texto
texto texto texto                         texto texto texto
```

```
\pagebreak
Primeira linha de texto \\
Segunda linha de texto \\
\vfill
Última linha de texto \\
\pagebreak
```

10 *CAPÍTULO 2. CRIANDO E ESTRUTURANDO UM DOCUMENTO*

Primeira linha de texto

Segunda linha de texto

Última linha de texto

Como você deve ter notado, o comando `\pagebreak` introduz uma quebra de página em um ponto específico do documento.

É importante ter em mente desde já que o input quase sempre conterá mais do que apenas texto. Haverá também *comandos*, que são instruções para controlar certos aspectos de formatação e a adição de símbolos especiais. `\pagebreak`, que acabamos ver, é um exemplo dentre vários que aparecerão daqui em diante. Como você notará, comandos são representados ou por certos caracteres especiais ou por palavras precedidas da barra invertida (`\`).

2.3.1 Comentários

Linhas iniciadas pelo símbolo de porcentagem (`%`) são ignoradas e não aparecem no output. Elas são consideradas “comentários” pelo compilador, isto é, instruções a serem ignoradas na compilação.

Eu vou aparecer.	Eu vou aparecer.
%Eu não.	

Essas linhas podem servir para o autor fazer comentários para ele mesmo no texto input. Naturalmente, esses comentários podem ser passados entre diferentes autores trabalhando em um mesmo texto, pois eles permitem dividir o texto impresso do texto “em construção”.

Todo número par é a soma de dois números primos.	Todo número par é a soma de dois números primos.
%Não tenho toda certeza disso..	

Da mesma forma, quando o símbolo de porcentagem aparece no meio de uma linha, tudo o que vem depois dele até o final da linha é ignorado.

Eu vou aparecer.%Eu não.	Eu vou aparecer. Mas eu vou.
Mas eu vou.	

2.3.2 Caracteres especiais

Existem alguns caracteres considerados especiais pelo compilador. Eles indicam que as expressões afetadas por eles são comandos, parâmetros de comandos, comentários, entre outros. Os principais são:

$$\$ \& \% \# _ \{ \}$$

Assim, para que eles apareçam no output como texto normal, deve-se digitá-los precedidos do símbolo `\`

$$\$ = \backslash\$ \& = \backslash\& \% = \backslash\% \# = \backslash\# _ = \backslash_ \{ = \backslash\{ \} = \backslash\}$$

2.4 Português

Se você estiver escrevendo um texto em português (brasileiro), não deixe de declarar no preâmbulo do documento o uso do pacote **babel** com a opção **brazil**:

```
\usepackage[brazil]{babel}
```

Esse pacote permite a separação de sílabas segundo as regras do português, além de converter certos títulos para o português. Assim, ao utilizar por exemplo o ambiente **abstract**, o resultado do título correspondente aparecerá automaticamente como **Resumo** e não **Abstract**.

A fim também de evitar problemas com a interpretação pelo Latex do uso de acentos, cedilhas, etc (inexistentes em inglês, língua para a qual o sistema foi originalmente pensado), declare no preâmbulo do documento o seguinte:

```
\usepackage[latin1]{inputenc}
```

2.5 Fontes

O tipo de fonte padrão do Latex é **Computer Modern**, mas outros tipos podem ser utilizados. Apenas a título de ilustração, há também opções como Times Roman e Palatino. Para que seu documento seja gerado com essas fontes, declare o uso de um dos pacotes abaixo no preâmbulo.

```
\usepackage{mathptmx} %para Times Roman
```

```
\usepackage{mathpazo} % para Palatino
```

Quanto ao tamanho das fontes, já vimos que o padrão do Latex é fonte tamanho 10 (10pt). As duas outras opções de tamanho são 11pt e 12pt. Como já mencionamos, essas opções devem ser declaradas na especificação de classe do documento, como no exemplo abaixo:

```
\documentclass[11pt]{article}
```

Para obter porções de texto em negrito, itálico ou sublinhado, utilize os comandos abaixo

<code>\textbf{negrito}</code>	negrito
<code>\textit{itálico}</code>	<i>itálico</i>
<code>\underline{sublinhado}</code>	<u>sublinhado</u>
<code>\textsc{caixa alta}</code>	CAIXA ALTA

Esses comandos podem ser encaixados um no outro:

<code>\textbf{\textit{negrito e itálico}}</code>	<i>negrito e itálico</i>
--	---------------------------------

Outros efeitos podem ser obtidos utilizando-se o pacote pacote [ulem](#) devidamente declarado no preâmbulo:

<code>\usepackage{ulem}</code> <code>...</code> <code>\uuline{duplo sublinhado}</code> <code>\uwave{ondulado}</code> <code>\sout{riscado}</code> <code>\xout{rasurado}</code>	<u>duplo sublinhado</u> ondulado riscado rasurado
--	---

2.6 Notas de Rodapé

Notas de rodapé são inseridas através do comando `\footnote{}` declarado no lugar em que o número da nota aparecerá. A nota propriamente dita será inserida automaticamente no rodapé da página em questão.

<code>Vamos testar\footnote{uma nota}</code> <code>pra ver\footnote{outra nota}</code> <code>o que vai acontecer.</code>	Vamos testar ¹ pra ver ² o que vai acontecer. <hr style="width: 20%; margin-left: 0;"/> ¹ uma nota ² outra nota
--	--

Se ao invés de notas de rodapé (footnotes), você preferir ou precisar agrupar as notas no final do texto (endnotes), declare o pacote **endnotes** no preâmbulo, seguido da especificação ilustrada no exemplo abaixo. Essa especificação insere o título **Notas** antes de listar as notas. Se preferir outro título, modifique esta especificação. Insira as notas da maneira usual, apenas substituindo `\footnote{}` por `\endnote{}` e utilize o comando **theendnotes** no local em que você deseja que as notas apareçam. Notas de rodapé e notas de fim podem co-existir em um mesmo documento.

<code>\usepackage{endnotes}</code> <code>\renewcommand{\notesname}</code> <code> {Notas}</code> <code>Vamos testar\endnote{uma nota}</code> <code>pra ver\endnote{outra nota}</code> <code>o que vai acontecer.</code> <code>\theendnotes</code>	Vamos testar ¹ pra ver ² o que vai acontecer. Notas ¹ uma nota ² outra nota
---	--

2.7 Título, Autor e Data

É comum indicar no início de um texto, seu título, autor(es) e data. Para tanto existem comandos específicos.

```
\title{<título>}
\author{<autor>}
\date{<data>}
```

Esses comandos devem ser declarados no preâmbulo do documento. Já o local de sua inserção no texto a ser produzido (geralmente no início) deve ser indicado no corpo do documento pelo comando `\maketitle`:

```
\documentclass{article}

\title{Samba, Futebol e Cerveja}
\author{João da Silva}
\date{2 de Fevereiro de 2009}

\begin{document}

\maketitle

Neste artigo, ...
```

Se houver mais de um autor, seus nomes devem ser separados pelo comando `\and`

```
\author{Jean da Silva \and José da Silva}
```

Se você quiser que título, autor e data apareçam em uma página separada (folha de rosto), inclua a especificação `titlepage` na especificação de classe:

```
\documentclass[titlepage]{article}
```

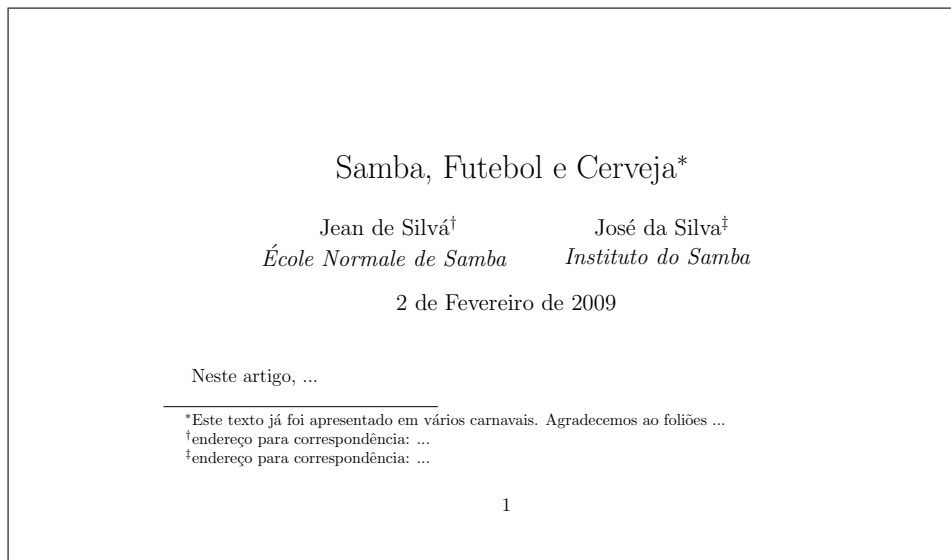
Caso você queira incluir a afiliação do autor debaixo de seu nome, utilize o comando `\\` logo após seu nome. (Esse comando nada mais é do que uma indicação de mudança de linha)

```
\author{Jean da Silva\\ École Normale de Samba
\and José da Silva\\ Instituto do Samba}
```

Caso você não queira que a data apareça, não ponha nada entre as chaves do comando (`\date{}`). Para gerar automaticamente a data do dia da produção do documento, utilize `\date{\today}`.

Notas de rodapé especiais (agradecimento, por exemplo) podem ser inseridas junto ao título, autores e data através do comando `\thanks{}`. Assim, o input abaixo geraria o output mostrado logo a seguir:

```
\title{Samba, Futebol e Cerveja\thanks{Este texto já foi
apresentado em vários carnavais. Agradecemos ao foliões}}
\author{Jean de Silv\thanks{Endereo: ...} \
\textit{cole Normale de Samba}
and Jos da Silva\thanks{Endereo: ...} \
\textit{Instituto do Samba}}
\date{2 de Fevereiro de 2009}
```



2.8 Numerao de pgina

O padro do Latex para a classe **article**  numerar as pginas com algarismos arbicos (1, 2, 3 ...) no centro do rodap de todas as pginas do documento.

Voc pode alterar esse padro globalmente, declarando no prembulo do documento o comando `\pagenumbering{}`. Entre chaves coloca-se o estilo desejado, que pode ser **arabic** para algarismos arbicos, **roman** para algarismos romanos, **alph** para letras minsculas, ou ainda **Alph** para letras maisculas.

Se for necessrio utilizar mais de um estilo de numerao no mesmo documento, os comandos acima podem ser utilizados mais de uma vez no corpo do documento. Assim, se seu documento se inicia com um prefcio

cujas páginas você deseja numerar com algarismos romanos e ao término do qual você deseja reverter para algarismos arábicos, utilize algo como o seguinte:

```
...
\begin{document}
\pagenumbering{roman}
...
<fim do prefácio>
\section{Introdução}
\pagenumbering{arabic}
...
\end{document}
```

Por fim, se você quiser que as páginas de seu documento não sejam numeradas, declare no preâmbulo `\pagestyle{empty}`. Caso, você queira que apenas uma determinada página (por exemplo, a primeira) não seja numerada, use o comando `\thispagestyle{empty}`

2.9 Espaçamento entre linhas

O padrão do Latex é o espaçamento simples entre linhas. Para obter espaçamento duplo ou espaçamento um e meio, declare o pacote **setspace** no preâmbulo, seguido do comando `\doublespacing` ou `\onehalfspacing`.

```
\usepackage{setspace} \doublespacing %para espaçamento duplo
```

Ou então:

```
\usepackage{setspace} \onehalfspacing %para espaçamento 1.5
```

Quaisquer outros valores para os espaçamentos são também possíveis através do comando `\setstretch{x}`, onde x é um número positivo inteiro ou fracionário representando o espaçamento. Exemplos: `\setstretch{0.7}`, `\setstretch{1.8}`.

2.10 Divisão do texto

2.10.1 Seções, subseções, ...

A divisão de um texto em capítulos, seções, subseções, etc é feita com os seguintes comandos:

```
\part \chapter
\section \subsection \subsubsection
\paragraph \subparagraph
```

No caso de a classe do documento ser **artigo**, a hierarquia começa em `\section`.

<code>\section{Metodologia}</code>		1 Metodologia
...		...
<code>\subsection{Experimentos}</code>		1.1 Experimentos
...		...

Essas divisões são por default numeradas automaticamente. Se você quiser gerar algo semelhante, mas sem número, use um asterisco no comando:

<code>\section*{Endereços}</code>		Endereços
-----------------------------------	--	------------------

2.10.2 Abstracts

Artigos acadêmicos costumam ter um resumo (abstract) no início. Para tanto, existe o ambiente **abstract**:

<code>\begin{abstract}</code>		Abstract
Neste artigo, mostro como é possível ganhar muito e trabalhar pouco, desde que se esteja disposto a abrir mão de certos princípios.		Neste artigo, mostro como é possível ganhar muito e trabalhar pouco, desde que se esteja disposto a abrir mão de certos princípios.
<code>\end{abstract}</code>		

2.10.3 Apêndice

Apêndices podem ser introduzidos da seguinte forma: primeiro, declara-se o comando `\appendix`, que tem por função re-formatar o contador de seções, que passará a usar letras maiúsculas (**A,B,C,...**) como padrão. Depois, basta usar o comando seção da maneira usual.

bla bla bla		bla bla bla
<code>\appendix</code>		A Demonstrações
<code>\section{Demonstrações}</code>		

2.11 Gerando um Sumário

Gerar um sumário (*table of contents*) é extremamente simples. O Latex faz tudo automaticamente, bastando que no corpo do texto seja declarado o

comando `\tableofcontents` exatamente no ponto em que você deseja que o sumário apareça. Todas as divisões numeradas (seções, subseções, etc.) serão incluídas no sumário com os respectivos números de páginas.

Caso você deseje que uma indicação referente a qualquer outro material apareça no sumário (por exemplo, os endereços para correspondência dos autores ao final de um artigo), você pode fazê-lo declarando no local em que este material aparece o seguinte comando:

```
\addcontentsline{toc}{<nível>}{<texto>}
```

Neste comando, *<nível>* indica o nível na hierarquia do sumário que o *<texto>* deve aparecer (section, subsection, etc.). No exemplo abaixo, o texto **Endereço para Correspondência** irá aparecer no mesmo nível que os títulos de seção.

```
\addcontentsline{toc}{section}{Endereço para Correspondência}
```

Esse tipo de comando pode ser usado também para fazer com que a seção de referências bibliográficas, que geralmente não é numerada, apareça no sumário.

```
\addcontentsline{toc}{section}{Referências}
```

Neste caso, a declaração acima deve vir junto ao comando `\bibliography{}`, responsável pela geração da lista bibliográfica, e que será discutido posteriormente.

É importante notar que para que o sumário seja inserido de fato no output, é necessário que o documento seja processado duas vezes. Na primeira rodada, ao notar o comando `\tableofcontents` no corpo do documento, o Latex abrirá um novo arquivo com o mesmo nome do arquivo original, mas com a extensão **.toc**. Todas as divisões numeradas (seções, subseções, etc) presentes no documento serão inseridas neste novo arquivo. Na segunda rodada, o Latex irá inserir o conteúdo deste arquivo **.toc** no output correspondente ao documento.

Capítulo 3

Ambientes

3.1 Listas

Há três tipos básicos de listas, os quais podem ser produzidos com os ambientes abaixo:

```
\begin{itemize}
  <itens com "bullets">
\end{itemize}
```

```
\begin{enumerate}
<itens numerados>
\end{enumerate}
```

```
\begin{description}
<itens rotulados>
\end{description}
```

Aqui está um exemplo com **itemize**:

```
\begin{itemize}
  \item primeiro item
  \item segundo item
  \item terceiro item
\end{itemize}
```

- primeiro item
- segundo item
- terceiro item

Se ao invés de ‘bullets’, você quiser um outro símbolo (+, por exemplo), indique-o da seguinte forma:

<code>\begin{itemize}</code>	+ primeiro item
<code>\item[+] primeiro item</code>	
<code>\item[+] segundo item</code>	+ segundo item
<code>\item[+] terceiro item</code>	
<code>\end{itemize}</code>	+ terceiro item

Agora um exemplo com **enumerate**:

<code>\begin{enumerate}</code>	1. primeiro item
<code>\item primeiro item</code>	
<code>\item segundo item</code>	2. segundo item
<code>\item terceiro item</code>	
<code>\end{enumerate}</code>	3. terceiro item

Aqui também você pode alterar a numeração. Note entretanto que este é um método braçal e que a numeração (i, ii, ...) não prosseguirá automaticamente:

<code>\begin{enumerate}</code>	i primeiro item
<code>\item[i] primeiro item</code>	
<code>\item[ii] segundo item</code>	ii segundo item
<code>\item[iii] terceiro item</code>	
<code>\end{enumerate}</code>	iii terceiro item

Por fim, um exemplo com **description**:

<code>\begin{description}</code>	luz primeiro item
<code>\item[luz] primeiro item</code>	
<code>\item[câmera] segundo item</code>	câmera segundo item
<code>\item[ação] terceiro item</code>	
<code>\end{description}</code>	ação terceiro item

Este tipo de lista pode ser útil na transcrição de entrevistas por exemplo, em que os rótulos dos itens podem se referir aos interlocutores em questão:

<code>\begin{description}</code>	Interlocutor 1 Como você reagiu?
<code>\item[Interlocutor 1]</code>	
<code> Como você reagiu?</code>	Interlocutor 2 Permaneci calado.
<code>\item[Interlocutor 2]</code>	
<code> Permaneci calado.</code>	Interlocutor 1 Humm ...
<code>\item[Interlocutor 1]</code>	
<code> Hummm ...</code>	
<code>\end{description}</code>	

Listas podem ser encaixadas.

<pre>\begin{itemize} \item bla bla \begin{enumerate} \item bla bla \item bla bla \end{enumerate} \item bla bla \end{itemize}</pre>	<ul style="list-style-type: none"> • bla bla <li style="margin-left: 2em;">1. bla bla <li style="margin-left: 2em;">2. bla bla • bla bla
--	--

Há outras duas maneiras de enumerar itens, ambas necessitando do pacote `\usepackage{paralist}`, que deve ser inserido no preâmbulo do documento.

Para listas destacadas do corpo do documento, tem-se o ambiente **compactenum**:

<pre>As quatro estações do ano são: \begin{compactenum}[(i)] \item primavera \item verão \item outono \item inverno \end{compactenum}</pre>	<pre>As quatro estações do ano são: (i) primavera (ii) verão (iii) outono (iv) inverno</pre>
---	--

Para listas no interior do documento, o ambiente **inparaenum**:

<pre>As quatro estações do ano são: \begin{inparaenum}[(a)] \item primavera, \item verão, \item outono, \item inverno. \end{inparaenum}</pre>	<pre>As quatro estações do ano são: (a) primavera, (b) verão (c) outono, (d) inverno.</pre>
---	---

Nos dois exemplos acima, com os comandos **compactenum** e **inparaenum**, podemos notar que basta que o tipo de enumeração — **(a)**, **(b)**, **(c)** ou **(i)**, **(ii)**, **(iii)** etc. — esteja especificado, entre colchetes, em seguida ao comando inicial — a saber, depois de `\begin{compactenum}` — para que cada entrada de lista (indicada pelo comando `\item`) apareça ordenada de acordo com o tipo de enumeração escolhido.

3.2 Tabelas

Tabelas são construídas através do ambiente **tabular**. Eis um exemplo que servirá de base para os comentários a seguir:

```

\begin{tabular}{|c|r||l|}
\hline
janeiro & fevereiro & março\\
\hline\hline
abril & maio & junho\\
\hline
julho & agosto & setembro\\
outubro & novembro & dezembro\\
\hline
\end{tabular}

```

janeiro	fevereiro	março
abril	maio	junho
julho	agosto	setembro
outubro	novembro	dezembro

Note que o número e a formatação das colunas é especificado logo na declaração de início do ambiente. A especificação `{|c|r||l|}` informa que serão 3 colunas, a primeira alinhada ao centro (c), a segunda à direita (r) e a terceira à esquerda (l). Informa ainda que a primeira e a segunda colunas serão separadas por uma linha vertical (|) e que a segunda e a terceira colunas serão separadas por duas linhas verticais (||). Note que haverá também linhas verticais nas bordas da tabela.

Note agora que a separação entre as células de uma mesma linha da tabela é feita através do comando `&`, e que o final de uma linha é indicado pelo comando `\\`.

Por fim, note que o comando `\hline` desenha uma linha horizontal entre duas linhas da tabela e `\hline\hline` insere duas linhas horizontais. Não havendo tais comandos, nenhuma linha visível é inserida.

Pode-se agrupar duas ou mais células de uma mesma linha através do comando `\multicolumn` como exemplificado abaixo:

```

\begin{tabular}{|c|c|c|}
\hline
\multicolumn{2}{|c|}{janeiro-fevereiro} & março \\
\hline
abril & maio & junho \\
\hline
julho & agosto & setembro \\
\hline
outubro & novembro & dezembro \\
\hline
\end{tabular}

```

janeiro-fevereiro		março
abril	maio	junho
julho	agosto	setembro
outubro	novembro	dezembro

Neste exemplo, o comando `\multicolumn{2}{|c|}{janeiro-fevereiro}` indica que as duas próximas células devem ser agrupadas em uma mesma coluna que deve ser centralizada e que tem por conteúdo ‘janeiro-fevereiro’.

Pode-se também agrupar duas ou mais células de uma mesma coluna através do comando `\multirow`. Isso, entretanto, requer o uso do pacote **multirow**, que deve ser declarado no preâmbulo. Segue um exemplo:

```
\begin{tabular}{|1|1||1|1|}
\hline
\multirow{3}{25mm}{\textbf{Primeiro Trimestre}} & janeiro &
\multirow{3}{25mm}{\textbf{Segundo Trimestre}} & abril \\
& fevereiro & & maio \\
& março & & junho \\
\hline
\multirow{3}{25mm}{\textbf{Terceiro Trimestre}} & julho &
\multirow{3}{25mm}{\textbf{Quarto Trimestre}} & outubro \\
& agosto & & novembro \\
& setembro & & dezembro \\
\hline
\end{tabular}
```

Primeiro Trimestre	janeiro fevereiro março	Segundo Trimestre	abril maio junho
Terceiro Trimestre	julho agosto setembro	Quarto Trimestre	outubro novembro dezembro

Neste exemplo, `\multirow{3}{25mm}{\textbf{Primeiro Trimestre}}` indica que as 3 próximas células da coluna em questão devem ser agrupadas, que a largura da coluna é de 25mm e que o texto correspondente é ‘**Primeiro Trimestre**’. Todos esses argumentos são obrigatórios.

O uso de **multirow** às vezes leva à necessidade de que traços horizontais que separam duas linhas de uma tabela não se estendam por todas as células destas linhas. Neste caso, ao invés de `\hline` utiliza-se `\cline{m-n}`, onde **m** e **n** especificam o intervalo em que o traço horizontal se estende. isto está ilustrado abaixo:

```
\begin{tabular}{|c|c|c|}
\hline
\multirow{3}{25mm}{\textbf{Primeiro Semestre}} & janeiro &
& fevereiro \\
& março & abril \\
& maio & junho \\
\hline
\end{tabular}
```

`\end{tabular}`

Primeiro Semestre	janeiro	fevereiro
	março	abril
	maio	junho

O ambiente `tabular` pode ainda se utilizado para inserir elementos na página, destacados do corpo do texto (dentro do ambiente de **citação** — `quote`), sem que haja colunas ou linhas entre esses elementos. Exemplo:

<pre>A seleção brasileira, nos últimos jogos, obteve os seguintes resultados: \begin{quote} \begin{tabular}{lcccl} Brasil & 3 & vs & 1 & Peru \\ \\ Brasil & 4 & vs & 2 & Chile \\ \end{tabular} \end{quote}</pre>	<pre>A seleção brasileira, nos últimos jo- gos, obteve os seguintes resultados: Brasil 3 vs 1 Peru Brasil 4 vs 2 Chile</pre>
--	---

3.3 Figuras

A inserção de figuras num texto é feita de modo bastante simples. Primeiro, o *arquivo contendo a imagem* tem de estar *na mesma pasta* onde se encontra o *arquivo da programação*. Depois, basta inserir no preâmbulo o pacote `\usepackage{graphicx}`, e utilizar o comando `\includegraphics{nome do arquivo com a respectiva extensão}` para inserir a imagem no corpo de texto:

Uma pintura muito comentada ao longo do tempo é Mona Lisa, de Leonardo da Vinci:\\

```
\includegraphics{mona.jpg}\\
```

O quadro Mona Lisa, também conhecido como La Gioconda, foi pintado entre os anos de 1503 e 1507.

Uma pintura muito comentada ao longo do tempo é Mona Lisa, de Leonardo da Vinci:



O quadro Mona Lisa, também conhecido como La Gioconda, foi pintado entre os anos de 1503 e 1507.

No exemplo acima, o quadro *Mona Lisa*, de Da Vinci, encontra-se num arquivo nomeado de **mona** e salvo na extensão `.jpg`. (Não se esqueça de que o arquivo **mona.jpg** se encontra na mesma pasta que o arquivo da programação.) Ao compilar o arquivo da programação, o TeXnicCenter procura o arquivo que, na programação, está indicado pelo comando `\includegraphics{}` — no caso, **mona.jpg** — para inserir no texto final a imagem (da Mona Lisa) quando for feita a impressão em pdf ou quando o texto for visualizado. Vê-se, portanto, que se trata de um modo bastante simples de inserir a imagem num texto.

(Obs: Para que a imagem não fique grudada nos parágrafos que lhe são anterior e posterior, usamos o *comando de espaçamento* (representado por duas barras invertidas `\\`) em seguida aos dois-pontos (em `Leonardo Da Vinci:\\`) e ao comando `\includegraphics{mona.jpg}`.

Mas há alguns detalhes importantes a respeito da inserção de figuras, que apresentaremos nas subseções a seguir.

3.3.1 Alterando o tamanho da figura

No exemplo anterior, reduzimos a imagem da Mona Lisa em 60% com relação ao tamanho da imagem que tínhamos originalmente em nosso arquivo. Se não o tivéssemos feito, muito provavelmente a imagem ocuparia grande parte da extensão do papel, o que seria indesejável. O problema é resolvido de modo simples, também. Para alterar o tamanho da figura, basta inserir a **opção** `[scale=PorcentagemDoOriginal]` depois do comando `\includegraphics` e antes do nome do arquivo, no caso, `{mona.jpg}`, da seguinte maneira:

```
\includegraphics[scale=0.2]
{mona.jpg}
```



Utilizando `\includegraphics` [opção de alteração]{nome do arquivo} podemos realizar várias alterações no tamanho da imagem. Vejamos três exemplos abaixo, em que a imagem da Mona Lisa teve seu tamanho alterado em três escalas diferentes, 0.15 (para a imagem à esquerda), 0.1 (para a imagem no centro) e 0.06 (para a imagem à direita):

```
(imagem à esquerda):
\includegraphics[scale=0.15]
{mona.jpg}

(imagem no centro):
\includegraphics[scale=0.1]
{mona.jpg}

(imagem à direita):
\includegraphics[scale=0.06]
{mona.jpg}
```



Até aqui, apresentamos um modo muito simplificado de inserir imagens no texto. A seguir apresentaremos algumas maneiras mais sofisticadas, com recursos mais eficientes.

3.3.2 Usando o ambiente “figure”

Uma maneira mais eficiente de inserir figuras é utilizando o ambiente **figure**. Nesse ambiente é possível inserir legendas nas figuras. Nesse caso, é necessário inserir no preâmbulo do documento os pacotes `\usepackage{graphicx}` e `\usepackage{caption}`. O primeiro é necessário para se criar o ambiente **figure**, onde são inseridas a figura e a legenda, enquanto o segundo é necessário para fazer pequenos ajustes na legenda.

A seguir apresentaremos detalhadamente o uso dos comandos necessários para inserir de modo simples e eficaz uma imagem no texto.

Para inserir uma imagem no ambiente **figure**, o comando que a introduz, a saber, `\includegraphics` deve estar incluído num bloco delimitado por `\begin{figure}` [especificador de posição] e `\end{figure}`. Vejamos um exemplo:


```
\begin{figure}[H]\centering
\includegraphics[scale=0.7]
{monet.jpg}
\end{figure}
```



No exemplo anterior, utilizamos o especificador `[H]`, para que a figura ficasse posicionada exatamente no lugar em que foi indicada na programação, ou seja, depois do parágrafo que termina com “Vejam um exemplo:”. O uso do comando `\centering`, depois de `\begin{figure}[H]`, insere a figura na posição central da página. A seguir, apresentamos uma tabela com diferentes **especificadores de posição** e as explicações sobre seu funcionamento.

especificador	permissão
h	insere a imagem num local próximo do que foi indicado na programação.
t	insere a imagem no topo da página.
b	insere a imagem na parte inferior da página.
p	insere a imagem numa página separada.
!	usado para determinar o melhor lugar para se posicionar a imagem.
H	insere a imagem exatamente no lugar onde está indicada na programação. É necessário inserir no preâmbulo o pacote <code>\usepackage{float}</code> .

À exceção do especificador `H`, que é utilizado para inserir a figura no local tal qual apontado na programação, os outros especificadores permitem ao \LaTeX maior flexibilidade para inserir a figura na página. Nesse caso, o \LaTeX normalmente procura uma determinada posição que, de acordo com a *permissão do especificador*, favoreça uma distribuição mais equilibrada dos elementos que compõem a página.

É possível também usar conjuntamente mais de um especificador, por exemplo, `[htb]`. Antes de explorarmos os recursos que valorizam a natureza flutuante das figuras, vejamos como inserir as legendas.

3.3.3 Inserindo legendas nas figuras

Para inserir legenda numa figura que se encontra no ambiente `figure`, basta inserir o comando `\caption{inserir aqui a legenda}`. Vejamos dois exemplos a seguir:

```

\begin{figure}[H]\centering
\includegraphics[scale=0.7]
{monet.jpg}
\caption{\textit{Sur les
planches de Trouville}.
Claude Monet, 1882.}
\end{figure}

```



Figura 3.1: *Sur les planches de Trouville*. Claude Monet, 1882.

No exemplo abaixo, apresentamos o quadro *Sur les planches de Trouville*, do pintor francês Claude Monet, com a legenda contendo as informações necessárias para esse tipo de referência.

```

\begin{figure}[H]\centering
\includegraphics[scale=0.7]
{renoir.jpg}
\caption{Pierre-Auguste
Renoir. 1881.
\textit{Doge's Palace}. Óleo
sobre tela, 54,5 x 65 cm.}
\end{figure}

```



Figura 3.2: Pierre-Auguste Renoir. 1881. *Doge's Palace*. Óleo sobre tela, 54,5 x 65 cm.

Conforme dissemos anteriormente, o pacote `\usepackage{caption}` permite realizar algumas alterações nas legendas. Vejamos alguns comandos utilizados, no exemplo acima, para personalizar as *legendas* das Figuras 1 e 2.

- I. Para que o **rótulo** (**Figura 1**, **Figura 2**, **Figura 3** etc.) de todas as legendas fique em negrito e para que todas as legendas se mantenham centralizadas abaixo da figura, é preciso inserir, no preâmbulo do documento e na linha abaixo do `\usepackage{caption}`, o comando `\captionsetup{justification=centering,labelfont=bf}`.

Com a **opção** `justification=centering` centralizamos a legenda, e com a **opção** `labelfont=bf`, inserimos o negrito nos **rótulos** (**Figura 1**, **Figura 2**, **Figura 3** etc.) das figuras.

- II. No caso da **Figura 3.2**, em que a legenda ocupa mais de uma linha, foi necessário que determinássemos uma quebra de linha entre `\textit{Doge's Palace.}` e `Óleo sobre tela`, com o comando das

duas barras invertidas `\`, para que pudéssemos redistribuir as informações das legendas.

A legenda também pode ser inserida acima da figura, bastando apenas que o comando `\caption{legenda}` seja introduzido, na programação, antes do comando `\includegraphics`. Vejamos a **Figura 3.3**:

```
\begin{picture}[H]
\caption{Claude Lorrain.
\textit{The Disembarkation
of Cleopatra at Tarsus}},
1918. Óleo sobre tela,
119 x 170 cm.}
\centering
\includegraphics[scale=1]
{lorrain.jpg}
\end{picture}
```

Figura 3.3: Claude Lorrain. *The Disembarkation of Cleopatra at Tarsus*, 1918. Óleo sobre tela, 119 x 170 cm.



3.3.4 Inserindo mais de uma figura

Quando se insere mais de uma figura na mesma página, é necessário, em muitos casos, que cada figura tenha uma legenda própria e, também, que todo o conjunto tenha uma única legenda. Para isso, é necessário o pacote `\usepackage{subfigure}`, que deve ser inserido no preâmbulo do documento. Vejamos um exemplo com duas figuras:

```
\begin{figure}[H]
\centering
\subfigure[Montagmorgen]
{\includegraphics[width=
0.3\textwidth]
{baluschek2.jpg}}
\subfigure[Großstadtwinkel]
{\includegraphics[width=
0.4\textwidth]
{baluschek.jpg}}
\caption{Hans Baluschek
(1870--1935)}
\end{figure}
```



(a) Montagmorgen



(b) Großstadtwinkel

Figura 3.4: Hans Baluschek
(1870–1935)

Vejam que é o comando `\subfigure` que insere o título das **sublegendas**, respectivamente **Montagmorgen** e **Großstadtwinkel**, no espaço entre colchetes “[]” que segue o comando `\subfigure`. Aqui é preciso chamar a atenção para um detalhe importante: o campo da sublegenda, indicado pelo comando `\subfigure`, não aceita nem acentos nem cedilha. Por isso, caso seja necessário inserir aí uma palavra que possua acento, deve-se utilizar o código do acento em *quetão*. Por exemplo, caso a sublegenda fosse a palavra **Composição**, deveríamos escrevê-la da seguinte maneira: `\subfigure[Composi\c c\~ao]`.

A seguir, apresentamos uma tabela com os códigos para os tipos mais usuais de acentuação:



Tipo	Visualização
<code>\'a</code>	à
<code>\~o</code>	ô
<code>\~a</code>	ã
<code>\'e</code>	é
<code>\c c</code>	ç

Para separar verticalmente uma imagem da outra, basta utilizar as duas barras invertidas `\\`. Caso quiséssemos separá-las horizontalmente, basta acrescentar os comandos `\,`, `\quad` ou `\qquad` (todos os quatro são comandos de espaçamento) entre cada comando `\subfigure`.

Tipo	Visualização
<code>a\,a</code>	a a
<code>a\quad a</code>	a a
<code>a\qquad a</code>	a a

3.3.5 Posição absoluta

Em certos usos específicos, pode ser útil determinar a *posição absoluta* dos elementos gráficos do texto. São dois os casos mais comuns de emprego de posições absolutas: quando se deseja imprimir além das margens do texto ou quando a distribuição dos elementos gráficos não segue o fluxo normal do texto (para criar sobreposições ou para uma especificação muito precisa da posição).

Uma das formas de se determinar posições absolutas é através do pacote **textpos**. Sua declaração é a seguinte:

```
{\usepackage[absolute]{textpos}}
```

O pacote habilita o ambiente `textblock*`, que cria uma caixa invisível na página de acordo com os parâmetros especificados. Os parâmetros são três: o tamanho horizontal do bloco e as coordenadas cartesianas x e y de seu canto superior esquerdo, a contar do canto superior esquerdo da folha de papel. Veja um exemplo:

```
\begin{textblock*}{20mm}(5mm,10mm)
  \includegraphics[scale=0.1]{mona.jpg}
\end{textblock*}
```

Estas instruções definem uma área de 20 milímetros de largura, com início no ponto a 5 mm da borda esquerda do papel (não da margem) e a 10 mm de sua borda superior. Nesse ponto preciso é inserida a figura.

Uma vez definida a área, qualquer elemento objeto pode ser localizado dentro dela, inclusive textos. Como ilustração, observe a área do rodapé desta página. São duas cadeias de textos sobrepostos, cujos comandos para produção são os seguintes:

```
\begin{textblock*}{10mm}(200mm,250mm)
000
\end{textblock*}
```

```
\begin{textblock*}{10mm}(201mm,250mm)
000
\end{textblock*}
```

Como se vê, os objetos podem ser sobrepostos, inclusive se são de tipos diferentes: texto sobre texto, imagem sobre imagem, texto sobre imagem. Por default, eles têm fundo transparente.

Uma opção útil do pacote é `showboxes`. Quando se deseja usá-la, deve-se declará-la juntamente com a opção `absolute`. Ela faz com que seja desenhada uma caixa demarcando os limites definidos, o que é útil para se ter uma noção mais clara do posicionamento, quando da criação do texto. Os comandos `\TPshowboxesfalse` e `\TPshowboxestrue` fazem sumir e aparecer a caixa, respectivamente.

3.4 Centralizar

```
\begin{center}
linha 1 \\ linha 2 \\ linha 3 ...
\end{center}
```

```
\begin{center}
Esse texto\\
vai aparecer\\
centralizado na página.
\end{center}
```

```
Esse texto
vai aparecer
centralizado na página.
```

3.5 Citações

Aqui está um texto normal, justificado e com certo afastamento das bordas da página.

```
\begin{quote}
Já esse texto aqui
vai estar mais afastado
das bordas da página.
\end{quote}
```

Aqui está outro texto normal, justificado e com certo afastamento das bordas da página.

Aqui está um texto normal, justificado e com certo afastamento das bordas da página.

Já esse texto aqui vai estar mais afastado das bordas da página.

Aqui está outro texto normal, justificado e com certo afastamento das bordas da página.

3.6 Epígrafes

O pacote **epigraph** é usado para formatar epígrafes. Deve ser declarado no preâmbulo do documento:

```
\usepackage{epigraph}
```

Seu uso é bem simples: o comando tem a forma `\epigraph{Citação}{Autor}`. Dois parâmetros podem ser (opcionalmente) ajustados *antes da chamada do comando* para o formato das epígrafes:

- `\setlength{\epigraphrule}{x pt}`. Ajusta para x pontos a espessura da linha que separa o nome do autor nas epígrafes. Caso prefira não ver a linha, ajuste o parâmetro para `0pt`.
- `\renewcommand{\epigraphsize}{\footnotesize}`. Ajusta o tamanho da fonte nas epígrafes. Nesse exemplo, os caracteres das epígrafes terão o mesmo tamanho daqueles das notas de rodapé.

Esses dois parâmetros podem ser modificados em qualquer ponto do texto, produzindo epígrafes de estilos variados. Esteticamente falando, porém, isso não é aconselhável.

```
\setlength{\epigraphrule}{0.1pt}
\renewcommand{\epigraphsize}{\footnotesize}
\epigraph{Le nez de Cléopâtre, s'il eût été plus court,
toute la face de la terre aurait changé.}
{Pascal, \textit{Pensées.}}
```

Le nez de Cléopâtre, s'il eût été
plus court, toute la face de la terre
aurait changé.

Pascal, *Pensées*.

3.7 Versos

O pacote `verse` divide e formata versos. Deve ser declarado no preâmbulo:

```
\usepackage{verse}
```

Trata-se de um pacote que oferece muitas opções de uso, que não se pretende esgotar aqui. Ele permite o uso do ambiente `verse`, marcado com um bloco `\begin{verse} / \end{verse}`. Como instruções gerais, deve-se usar os comandos:

- `\poemtitle*{Título}`, para colocar o título do poema. O símbolo é opcional e, se estiver presente, indica que o título não deve figurar no sumário.
- `\poemlines{n}`. Opcional. Numera as linhas do poema de n em n . Por default, os números aparecem à direita dos versos; caso queira vê-los à esquerda, acrescente em seguida o comando `\verselinenumbersleft`.
- `\flagverse{Subtítulo}`. Opcional. Nomeia seções internas do poema. Deve ser usado no interior do ambiente `verse`.

Os versos do poema devem sempre terminar com `\\`. As estrofes são separadas por linhas em branco. Se a numeração de linhas for usada, deve-se terminar uma estrofe com o comando `\\!`. O símbolo `!` faz com que a linha em branco separadora das estrofes não seja contada na numeração. Veja-se o exemplo a seguir:

```
\poemtitle*{\textit{OS LUSÍADAS}}
\poemlines{5}
\verselinenumbersleft
```



```

\begin{verse}

\flagverse{\textsc{Canto I}}
As armas e os Barões assinalados\\
Que da Ocidental praia Lusitana\\
Por mares nunca de antes navegados\\
Passaram ainda além da Taprobana,\\
Em perigos e guerras esforçados\\
Mais do que prometia a força humana,\\
E entre gente remota edificaram\\
Novo Reino, que tanto sublimaram;\\!

E também as memórias gloriosas\\
Daqueles Reis que foram dilatando\\
A Fé, o Império, e as terras viciosas\\
De África e de Ásia andaram devastando,\\
E aqueles que por obras valerosas\\
Se vão da lei da Morte libertando,\\
Cantando espalharei por toda parte,\\
Se a tanto me ajudar o engenho e arte.\\

\end{verse}

```

OS LUSÍADAS

CANTO I	As armas e os Barões assinalados Que da Ocidental praia Lusitana Por mares nunca de antes navegados Passaram ainda além da Taprobana,
5	Em perigos e guerras esforçados Mais do que prometia a força humana, E entre gente remota edificaram Novo Reino, que tanto sublimaram;
10	E também as memórias gloriosas Daqueles Reis que foram dilatando A Fé, o Império, e as terras viciosas De África e de Ásia andaram devastando,
15	E aqueles que por obras valerosas Se vão da lei da Morte libertando, Cantando espalharei por toda parte, Se a tanto me ajudar o engenho e arte.

3.8 Cabeçalhos e rodapés

Para compor o cabeçalho e o rodapé do documento é preciso antes acrescentar, no preâmbulo do documento, `\usepackage{fancyhdr}`. Com este pacote podemos personalizar o *cabeçalho* e o *rodapé* do texto com informações do tipo: (a) Nome do autor, (b) Nome de revista, (c) Nome da instituição de filiação do autor e também (d) o número da página.

Como se trata de uma operação que deve valer para todo o documento, todos os comandos utilizados para gerar os *cabeçalhos* e os *rodapés* devem ser inseridos no preâmbulo, após as declarações de pacotes.

Os comandos utilizados são muito simples e bastante intuitivos. O cabeçalho (`head`) e o rodapé (`foot`) possuem cada um três campos — a saber, o direito (indicado pela letra **r** de right), o centro (por **c** de center) e o esquerdo (por **l** de left). Com isso, forma-se a tríade `\rhead`, `\chead` e `\lhead` para o cabeçalho, e a tríade `\rfoot`, `\cfoot` e `\lfoot` para o rodapé:

<code>\lhead</code>	<code>\chead</code>	<code>\rhead</code>
<hr/> Texto texto texto... Texto texto texto... Texto texto texto...		
<code>\lfoot</code>	<code>\cfoot</code>	<code>\rfoot</code>

A seguir apresentamos um exemplo de uso das ferramentas do `fancyhdr`:

<code>LaTeX</code> para linguistas	<i>por M. Ferreira, M. Lopes e F. Merçon</i>
<hr/> Para compor o cabeçalho e o rodapé do documento é preciso antes acrescentar, no preâmbulo do documento, <code>\usepackage{fancyhdr}</code> . Com este pacote podemos personalizar o <i>cabeçalho</i> e o <i>rodapé</i> do texto com informações do tipo: (a) Nome do autor, (b) Nome de revista, (c) Nome da instituição de filiação do autor e também (d) o número da página.	
<hr/> 14	

Veja a programação que usamos para compor o cabeçalho e o rodapé ilustrados acima:

```
\pagestyle{fancy}
```

```
\lhead{\LaTeX para linguistas}
```

```
\chead{}
```

```
\rhead{\it por M. Ferreira, M. Lopes e F. Merçon}
```

```
\tfoot{}
\cfoot{\thepage}
\rfoot{}
```

Para que seja substituída a numeração de páginas automática do L^AT_EX, é só inserir, no campo desejado, o comando `\thepage`, como o fizemos no centro do rodapé, por meio do comando `\cfoot{\thepage}`. Quanto ao comando `\pagestyle{fancy}`, é ele que faz com que as informações indicadas no cabeçalho e no rodapé sejam visualizadas. Mesmo não tendo sido preenchidos os campos `\thead`, `\tfoot` e `\rfoot`, eles não podem ser excluídos da programação.

Vejamos agora o exemplo abaixo:

<i>L^AT_EX</i> para linguistas	6 CABEÇALHOS E RODAPÉS
<hr/> <p>6 Cabeçalhos e rodapés</p> <p>Para compor o cabeçalho e o rodapé do documento é preciso antes acrescentar, no preâmbulo do documento, <code>\usepackage{fancyhdr}</code>. Com este pacote podemos personalizar o <i>cabeçalho</i> e o <i>rodapé</i> do texto com informações do tipo: (a) Nome do autor, (b) Nome de revista, (c) Nome da instituição de filiação do autor e também (d) o número da página.</p> <hr/> <p style="text-align: center;">14</p>	

No exemplo da página anterior, notamos que, no canto direito do cabeçalho, aparece o título **6 Cabeçalhos e rodapés** da seção da qual faz parte a página. Esse é um recurso que pode ser obtido com a inserção do comando `\leftmark` em qualquer um dos campos do cabeçalho ou do rodapé, no presente caso, empregamo-lo no campo `\rhead{\leftmark}`. Caso se queira inserir também os títulos das subseções, é inserir `\rightmark` em um dos campos, à maneira do que fizemos com o `\leftmark`. A seguir, um exemplo (**Exemplo 3**) com o uso dos dois recursos (em que `\leftmark` introduz o título da seção **4 Figuras** no canto esquerdo do cabeçalho, e o comando `\rightmark`, por sua vez, introduz o título da subseção **4.1 Alterando o tamanho da figura** no canto direito do cabeçalho. No rodapé, por sua vez, no campo central, introduzimos somente o número da página, com o uso do comando `\thepage`.

<p>4 FIGURAS</p> <hr/> <p>4 Figuras</p> <p>A inserção de figuras num texto é feita de modo bastante simples. Primeiro, o <i>arquivo</i> <i>contendo a imagem</i> tem de estar <i>na mesma pasta</i> onde se encontra o <i>arquivo</i> <i>a pro-</i> <i>gramação.</i></p> <p>[...]</p> <p>Mas há alguns detalhes importantes a respeito da inserção de figuras, que apresentare- mos nas subseções a seguir.</p> <hr/> <p style="text-align: center;">5</p>	<p>4.1 ALTERANDO O TAMANHO DA FIGURA</p>
<p>4.1 Alterando o tamanho da figura</p> <p>No exemplo anterior, reduzimos a imagem da Mona Lisa em 60% com relação ao tamanho da imagem que tínhamos originalmente em nosso arquivo. [...]</p> <hr/> <p style="text-align: center;">6</p>	

Veja abaixo como é bem simples a programação desse recurso de inserção de títulos de **seções** e de **subseções** no cabeçalho.

```
\pagestyle{fancy}

\lhead{\leftmark}
\chead{}
\rhead{\rightmark}

\lfoot{}
\cfoot{\thepage}
\rfoot{}
```

Uma última observação: para que o cabeçalho e o rodapé não apareçam numa determinada página — por exemplo, na primeira página do artigo —, basta introduzir na página em questão o comando `\thispagestyle{empty}`.

3.9 Cores

Há pelo menos duas maneiras muito simples de fazer uso de cores no \LaTeX . A seguir daremos uma demonstração de cada uma delas, em diferentes usos.

3.9.1 Cores básicas

O modo mais simples de fazer uso de cores é inserir o comando `{\color{ } }`, indicando, no campo vazio entre as chaves, o nome da cor desejada, e em seguida, o trecho em que se pretende aplicar a respectiva cor. Vejamos alguns exemplos:

Podemos usar `{\color{blue}` para que parte do texto fique azul}, mas veja que a parte que se encontra fora do trecho selecionado é mantida na cor padrão do \LaTeX , que é a preta.

Podemos usar `{\color{blue}` para que parte do texto fique azul}, mas veja que a parte que se encontra fora do trecho selecionado é mantida na cor padrão do \LaTeX , que é a preta.

Veja que os nomes das cores estão em inglês. Por exemplo,

Vermelho se faz com `{\color{red}Vermelho}`

Cinza se faz com `{\color{gray}Cinza}`

Verde se faz com `{\color{green}Verde}`

Azul se faz com `{\color{blue}Azul}`

Laranja se faz com `{\color{orange}Laranja}`

... e assim por diante.

3.9.2 Alterando cores

Podemos obter de duas maneiras cores diferentes das apresentadas anteriormente: (a) graduando o tom da cor escolhida, (b) combinando mais de uma cor com específicos valores para cada uma.

Graduando o tom da cor

Ao inserirmos uma opção de cor no comando `\color`, como por exemplo, `\color{blue}`, é como se estivéssemos determinando um valor de pigmentação equivalente a 100 % da cor escolhida. Se esse valor limite for alterado, obteremos diferentes tons e, com isso, aumentamos consideravelmente nossa “paleta” de cores. Para definir esse valor, basta acrescentar, ao lado do nome da cor, um ponto de exclamação ! e um número entre 0 e 100 (esse número equivale à porcentagem de pigmentação da cor selecionada). Vejamos alguns exemplos:

(a) <code>{\color{blue} 100\%}</code>	(a) 100%
(b) <code>{\color{blue!70} 70\%}</code>	(b) 70%
(c) <code>{\color{blue!40} 40\%}</code>	(c) 40%
(d) <code>{\color{blue!10} 10\%}</code>	(d) 10%

O mesmo pode ser feito com a cor preta. Nesse caso, obteremos diferentes tons de cinza:

(a) <code>{\color{black} 100\%}</code>	(a) 100 %
(b) <code>{\color{black!70} 70\%}</code>	(b) 70%
(c) <code>{\color{black!40} 40\%}</code>	(c) 40%
(d) <code>{\color{black!10} 10\%}</code>	(d) 10%

Note que, em ambos os casos, é como se estivéssemos misturando as cores escolhidas (no caso, o azul e o preto) com a cor branca. A seguir veremos como obter outras cores a partir da combinação de duas cores.

Combinando cores

Utilizando um recurso bem semelhante ao anterior, quando graduamos uma determinada cor, podemos combinar duas cores para, com isso, gerar uma outra cor. Basta escolher duas cores a serem combinadas e definir o quantidade de cada uma que gostaríamos de misturar na “paleta”. Por exemplo, para misturar o azul e o preto, com 60 % de azul e 40 % de preto,

basta fazer o seguinte: `{\color{blue!60!black}}`. Note que a cor situada à esquerda (o azul) é que recebe o valor de 60 %, enquanto a outra, situada à direita (o preto), recebe o restante que falta para que seja completado o total de 100 %, ou seja, 40 %.

Vejamos, a seguir, alguns exemplos de diferentes combinações de azul e preto:

(a) <code>{\color{blue!80!black}</code> 80\% azul, 20\% preto}	(a) 80% azul, 20% preto
(b) <code>{\color{blue!60!black}</code> 60\% azul, 40\% preto}	(b) 60% azul, 40% preto
(c) <code>{\color{blue!40!black}</code> 40\% azul, 60\% preto}	(c) 40% azul, 60% preto
(d) <code>{\color{blue!20!black}</code> 20\% azul, 80\% preto}	(d) 20% azul, 80% preto

Capítulo 4

Aplicações Linguísticas

4.1 Símbolos Fonéticos

4.1.1 Instalação

O pacote TIPA, para transcrições fonéticas, além de trazer um conjunto bastante completo dos símbolos do Alfabeto Fonético Internacional, possibilita recursos de orientação dos símbolos, posicionamento de diacríticos (incluindo combinações de dois ou mais deles) e marcação de tons. Inclui ainda símbolos usados nos estudos históricos do Indo-Europeu, na descrição específica de línguas do Extremo Oriente e diacríticos especiais para marcar a qualidade da produção da fala (em casos especiais, como as apraxias, por exemplo).

Esteticamente falando, uma das vantagens do resultado final é que as fontes fonéticas seguem o mesmo padrão de estilo do texto cursivo. As maiores vantagens, entretanto, são mesmo ligadas à exatidão do resultado: o texto ou *slide* feito com L^AT_EX e TIPA pode ser lido e editado em qualquer computador, sem depender de fontes de caracteres externas ou de versões específicas de softwares, diferentemente dos textos gerados por editores convencionais. Isso significa que, mesmo em computadores diferentes ou em tempos posteriores, não só o PDF gerado inicialmente com as fontes fonéticas como também o arquivo .TEX que terá servido de fonte permanecerão inalterados. Nada de trocar as transcrições fonéticas (ou qualquer outra parte do texto) por símbolos de Wingdings.

Para usar o pacote TIPA, este deve ser declarado no preâmbulo:

```
\usepackage{tipa}
```

Algumas opções úteis do pacote são: `[tone]`, se você pretende indicar marcações tonais, e `[extra]`, que permite o uso de diacríticos especiais para marcação da qualidade da produção da fala. Os pacotes podem ser declarados em conjunto, separados por vírgula, como neste exemplo:

```
\usepackage[tone,extra]{tipa}.
```

4.1.2 Transcrevendo

A transcrição é indicada por um comando `\textipa{}`. Entre as chaves, como parâmetro obrigatório, vêm os caracteres da transcrição. Em geral, eles são codificados como equivalentes mais ou menos próximos das letras do alfabeto (não-fonético). Seguem as representações dos sons do português, para começar:

Vogais orais do português

<code>\textipa{[i e E a 0 o u]}</code>	[i e ε a ɔ o u]
<code>\textipa{[I U 5]}</code>	[ɪ ʊ ɐ]

Repare que os símbolos idênticos à escrita alfabética são codificados da mesma forma; símbolos especiais (isto é, não-alfabéticos) têm codificações especiais. O uso de letras alfabéticas maiúsculas é bastante frequente, como no caso de ϵ e ɔ ; já no caso de ɐ , foi usado um número (5), pois o comando `\textipa{[A]}` produziria um [ɑ].

Como saber, então, qual é o símbolo produzido em cada caso? Vamos listar aqui o conjunto dos símbolos das vogais e das consoantes mais usuais do Alfabeto Fonético para o português; no caso de necessidades mais especiais e de usos avançados, o [manual](#) do pacote TIPA traz listas completas em seus apêndices.

Por ora, aqui está outra estratégia para gerar símbolos. Trata-se de descrições dos símbolos usadas em comandos.

<code>\textipa{\textschwa}</code>	ə
<code>\textipa{\textfishhookr}</code>	ɾ
<i>Giro:</i> <code>\textipa{\textturna}</code>	ɞ
<i>Reverso:</i> <code>\textipa{\textreve}</code>	ɘ
<i>Inverso:</i> <code>\textipa{\textinvscr}</code>	ɸ

Os três últimos comandos descrevem “movimentos” dos símbolos: giro (*turn*), reverso (*rev*) e inverso (*inv*). Muitos símbolos podem ser gerados dessa maneira¹.

Vogais nasais

O acréscimo do diacrítico $\tilde{}$ às vogais é feito pela anteposição do comando `\~`.

¹As possibilidades combinatórias, no entanto, não são livres, mas restritas aos símbolos consagrados do Alfabeto Fonético. Caso você precise de mais, o L^AT_EX permite virtualmente qualquer manipulação de fontes e de posições e sentidos de escrita através de outros recursos, como o XeTeX.

<code>\textipaõ</code>		õ
<code>\textipa{\~textschwa}</code>		ẽ
<code>\textipa{\~5}</code>		ẽ

Consoantes do português

A maioria das consoantes em nossa língua não requer codificação especial, já que o símbolo alfabético coincide com o símbolo no IPA. Eis aqui uma lista de outros casos:

<code>\textipa{S}</code>		ʃ
<code>\textipa{Z}</code>		ʒ
<code>\textipa{L}</code>		ʎ
<code>\textipa{\textltailn}</code>		ɲ
<code>\textipa{N}</code>		ŋ
<code>\textipa{\textfishhookr}</code>		ɾ
<code>\textipa{\textturnrrtail}</code>		ɽ
<code>\textipa{[\textgamma]}</code>		ɣ
<code>\textipa{\t*{tS}}</code>		tʃ
<code>\textipa{\t*{dZ}}</code>		dʒ

Outros símbolos úteis:

<code>\textipa{\textprimstress}</code>		ˈ
<code>\textipa{\textsecstress}</code>		ˌ
<code>\textipa{:}</code>		ː
<code>\textipa{;}</code>		ˑ

4.1.3 Atalhos no teclado

De maneira geral, como aqui a transcrição não depende de ações de “inserir símbolos”, com a prática ela vai se tornando muito ágil, pois os símbolos mais usuais do Alfabeto Fonético são digitados com um só toque no teclado. Veja a tabela de equivalências abaixo, onde o código ASCII corresponde àquilo que você digita²:

4.1.4 Tabela completa dos símbolos IPA

A tabela *tipachart*, de autoria de Matt Makashay, disponível neste [link](#), apresenta todos os símbolos da versão *standard* (não-expandida) do Alfabeto Fonético Internacional. É importante salientar que essa tabela, por mais

²Essa tabela mostra apenas uma pequena parte dos símbolos que podem ser gerados com o pacote TIPA. Não seria viável elencar aqui todas as possibilidades, que corresponderiam a algumas centenas de símbolos.

<i>ASCII</i>	:	;	"							
<i>TIPA</i>	:	;	"							
<i>ASCII</i>	0	1	2	3	4	5	6	7	8	9
<i>TIPA</i>	u	i	ʌ	ɜ	ɥ	e	ɒ	ɣ	ə	ə
<i>ASCII</i>	@	A	B	C	D	E	F	G	H	I
<i>TIPA</i>	ə	ɑ	β	ɕ	ð	ɛ	ϕ	ɣ	ɦ	ɪ
<i>ASCII</i>	J	K	L	M	N	O	P	Q	R	S
<i>TIPA</i>	j	ɸ	ʌ	ɱ	ɲ	ɔ	ʔ	ʕ	ɾ	ʃ
<i>ASCII</i>	T	U	V	W	X	Y	Z			
<i>TIPA</i>	θ	ʊ	ʊ	ʍ	χ	ɣ	ʒ			

Tabela 4.1: Atalhos no teclado para o pacote TIPA

que seja suficiente para a maioria dos usos do alfabeto fonético, não apresenta todos os símbolos do pacote TIPA e não substitui a lista de símbolos apresentada no apêndice do manual. Trata-se assim mesmo de um excelente guia de referência rápida, que vale a pena ter sempre à mão quando se faz transcrições fonéticas.

4.1.5 Diacríticos

Os diacríticos são normalmente introduzidos pelo comando `\|` seguido pelo identificador do diacrítico em questão e do fone afetado por ele. Por exemplo,

<code>\textipa{\ [t}</code>		t̃
<code>\textipa{\ (a}</code>		ã
<code>\textipa{\ w{k}</code>		k̃

Combinações de diacríticos, tons e outros recursos

- Símbolos sobrescritos:

`t\textsuperscript h k\textsuperscript {wj}`: t^h k^{wj}

Esse não é um recurso específico do pacote TIPA, mas da tipografia em \LaTeX como conjunto, e está sendo lembrado aqui porque é de uso bastante comum em transcrições fonéticas. Uma possibilidade adicional dos sobrescritos é que eles podem ser recursivamente utilizados:

`t\textsuperscript{hj}`: t^{h^j}

- Símbolos superpostos:

`\tipaUpperaccent [.2ex]{\lower.8ex\hbox{\textipa{\super@}}}{a}`:
^aa

- Símbolos sobrepostos:

– “b” + “/”: `\ipaclap{\textipa{b}}{\textipa{/}}`: \mathbb{b} ;

– “I” + “=”: `\ipaclap{I}{\textipa{=}}`: \mathbb{I}

- Diacríticos combinados:

`\textipa{\'\{"{\u*{e}}}`: \acute{e}

- Marcação de tons:

`\tone{55} \tone{22} \tone{51}`: $\uparrow \downarrow \backslash$

4.2 Símbolos Lógico-Semânticos

Em Semântica Formal, usa-se com frequência colchetes duplos (`[[]]`) para a denotação/extensão de uma determinada expressão, como no exemplo abaixo:

`[[todo]]` = $\lambda f.\lambda g. \forall x : f(x) \rightarrow g(x)$

O uso desses colchetes requer a declaração do pacote **stmaryrd** no preâmbulo do documento. Os comandos para os colchetes são os seguintes:

<code>\lbracket</code>	<code>[[</code>
<code>\rrbracket</code>	<code>]]</code>

Esses comandos são considerados símbolos matemáticos e portanto devem ser usados no escopo de `$$`. Para facilitar o uso desses colchetes, insira no final do preâmbulo de seu documento a seguinte linha:

```
\newcommand{\den}[1]{\$ \lbracket $#1 \$ \rrbracket \$}
```

Com isso, cria-se um novo comando `\den{}`. Agora, basta você inserir dentro das chaves a expressão em questão. Por exemplo:

<code>\den{todo}</code> = ...	<code>[[todo]]</code> = ...
-------------------------------	-----------------------------

Quanto aos símbolos lógico-matemáticos, segue abaixo uma lista parcial contendo os de uso mais frequente. Uma vez mais, lembre-se de que esses comandos devem ser usados no escopo de `$$`.

<code>\land</code>		\wedge
<code>\lor</code>		\vee
<code>\rightarrow</code>		\rightarrow
<code>\neg</code>		\neg
<code>\forall</code>		\forall
<code>\exists</code>		\exists

Por fim, letras gregas também são consideradas símbolos matemáticos. Alguns exemplos:

<code>\alpha</code>		α
<code>\beta</code>		β
<code>\phi</code>		ϕ
<code>\psi</code>		ψ
<code>\iota</code>		ι
<code>\theta</code>		θ
<code>\lambda</code>		λ

Com isso em mente, você já tem condições de entender o código que gerou a denotação da palavra *todo*, que apresentamos mais acima:

```
\den{todo} =
  $\lambda f.\lambda g.\ \forall x: f(x) \rightarrow g(x)$
```

4.3 Símbolos utilizados em Semiótica

É comum, em textos de semiótica, o uso de diversos símbolos, seja para indicar direções, como \rightarrow ou \leftarrow , ou ainda para especificar alguns conceitos utilizados em análises. Por exemplo, para ilustrar simbolicamente que o sujeito está em *conjunção* ou *disjunção* com o *objeto-valor*, fazemos uso dos seguintes símbolos: **(a)** $S \cap O_v$ ou **(b)** $S \cup O_v$. Podemos representar também a fase de manipulação em que um *sujeito* leva outro *sujeito* a entrar em *conjunção* com determinado *objeto-valor*: **(c)** $S_1 \text{ (João)} \rightarrow [S_2 \text{ (Pedro)} \cap O_v \text{ (vida)}]$.

A edição desses símbolos, no L^AT_EX, é bastante simples.

Para editar os operadores de:	Empregamos os seguintes comandos:
conjunção \cap	<code>\$\$\cap\$</code>
disjunção \cup	<code>\$\$\cup\$</code>
transformação \rightarrow	<code>\$\$\rightarrow\$</code>

Caso se queira alongar o operador de transformação, use `$$\longrightarrow$`, e obterá: \longrightarrow . Para alterar a direção da seta, veja os exemplos abaixo, com seus respectivos comandos:

↑	<code>\uparrow</code>
↓	<code>\downarrow</code>
↕	<code>\updownarrow</code>
↔	<code>\leftrightarrow</code>
←	<code>\longleftarrow</code>
↔	<code>\longleftrightarrow</code>
↗	<code>\nearrow</code>
↘	<code>\searrow</code>
↙	<code>\swarrow</code>
↖	<code>\nwarrow</code>

Obs: em todos os exemplos apresentados acima, você deve ter notado que os comandos estão delimitados pelo símbolo \$. Lembre-se, seu uso, nesses casos, é imprescindível!

Quanto à letra ou texto que normalmente acompanham a notação das funções actanciais e que se posicionam *no pé* da letra — como em $D_{\text{estinator}}$ — é preciso antes de tudo adicionar, no preâmbulo do documento, o pacote `\usepackage{fixltx2e}`, e depois, basta utilizar o comando `\textsubscript{}`.

Vejam os exemplos:

- No mito de Prometeu, podemos verificar duas etapas do esquema narrativo canônico. Primeiro, na etapa da *performance*, Prometeu rouba o fogo de Zeus e entrega-o aos homens. Depois, na etapa da *sanção*, Zeus determina que Hefesto aprisione Prometeu pela ação. Os dois esquemas podem ser representados da seguinte maneira:

1. performance $S_{1 \text{ Prometeu}} \rightarrow [S_{2 \text{ homens}} \cap O_{\text{fogo}}]$.

– `S\textsubscript{1 Prometeu}`
`\rightarrow` `[S\textsubscript{2 homens} \cap`
`O\textsubscript{fogo}]`.

2. sanção $D_{\text{estinator Zeus}} \rightarrow D_{\text{estinatário Hefesto}} \rightarrow [S_{1 \text{ Prometeu}} \cup O_{\text{bjeto-valor vida}}]$.

– `D\textsubscript{estinator Zeus}`
`\rightarrow` `D\textsubscript{estinatário Hefesto}`
`\rightarrow` `[S\textsubscript{1 Prometeu} \cup`
`O\textsubscript{bjeto-valor vida}]`.

Nos exemplos acima, usamos o comando `\textsubscript{}` para posicionar um texto ou uma letra no pé de uma determinada função actancial, por exemplo, $S_{1 \text{ Prometeu}}$. Para adicionar um texto no topo de uma letra, como em $F_{\text{ranz}} K^{\text{afka}}$, é só usar o comando `\textsuperscript{}`, e inserir no campo, entre as chaves, o texto que se pretende inserir na altura da cabeça da letra. Como ilustração adicional, considere o exemplo abaixo, gerado a

partir do código mostrado a seguir:

```
estudos semióticos
```

```
\textsuperscript{es}t\textsubscript{u}do\textsubscript{s}
\textsubscript{se}m\textsuperscript{i}ót
\textsubscript{i}co\textsuperscript{s}
```

4.4 Outros símbolos

Caso você não encontre um símbolo qualquer de que esteja precisando, há duas saídas recomendadas. A primeira é usar o programa [Detexify](#), que procura identificar o símbolo de acordo com um desenho feito por você. A segunda é recorrer às lista de símbolos, nas versões [compacta](#) e [completa](#).

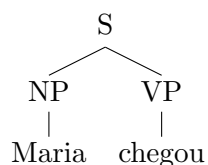
4.5 Árvores Sintáticas

Ilustraremos a criação de árvores sintáticas através do uso do pacote **qtree**, que deve ser declarado no preâmbulo:

```
\usepackage{qtree}
```

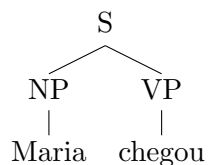
O comando para a geração das árvores é `\Tree`. A especificação da árvore é feita através da notação de colchetes rotulados, como no exemplo abaixo:

```
\Tree [.S [.NP Maria ] [.VP chegou ] ]
```



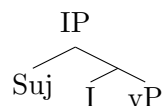
Note que os rótulos são inseridos logo após os colchetes, seguidos de um ponto. Esses rótulos tanto podem vir junto ao colchete de abertura ([]) quanto do colchete de fechamento (]) de um constituinte, de acordo com sua preferência. Assim, a árvore acima poderia também ser obtida da seguinte forma:

```
\Tree [ [ Maria ].NP [ chegou ].VP ] .S
```



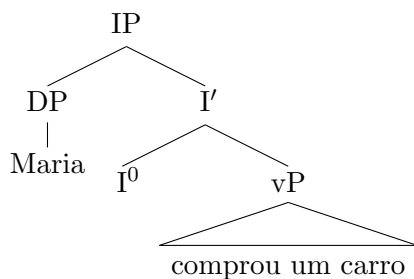
Pode-se também omitir um rótulo:

```
\Tree [.IP Suj [ I vP ] ]
```



Triângulos representando constituintes não analisados podem ser gerados com o comando `\qroof{}`:

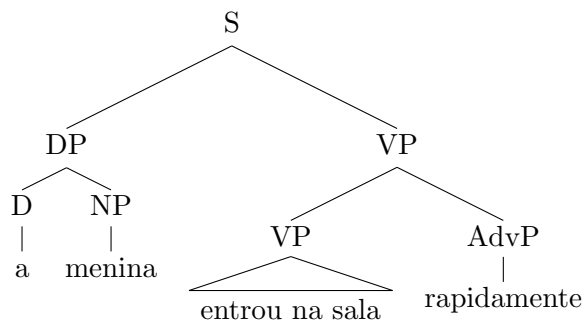
```
\Tree [.IP [.DP Maria ] [.I\1 I\0 \qroof{comprou um carro}.vP ]]
```



Note o uso dos comandos `\0` e `\1` para produzir I^0 e I' .

Aqui vai mais um exemplo para ilustrar:

```
\Tree [.S [.DP [.D a ] [.NP menina ] ] [.VP \qroof{entrou na sala}.VP [.AdvP rapidamente ] ] ]
```



É importante notar que o padrão do pacote **qtree** é gerar as árvores centralizadas nas páginas. Caso você queira que elas sejam geradas na posição normal do texto (como neste documento), declare no preâmbulo a opção `\qtrecenterfalse`, logo depois da declaração do próprio **qtree**:

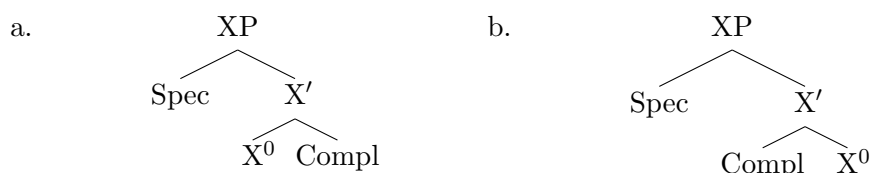
```
\usepackage{qtree}
\qtrecenterfalse
```

Dessa forma, você pode inclusive gerar árvores uma ao lado da outra:

```
\textit{Estruturas de Núcleo Inicial e de Núcleo Final}\
```

```
a.\ \Tree [.XP Spec [.X\1 X\0 Compl ] ]
\hspace*{1.3cm} b.\ \Tree [.XP Spec [.X\1 Compl X\0 ] ]
```

Estruturas de Núcleo Inicial e de Núcleo Final

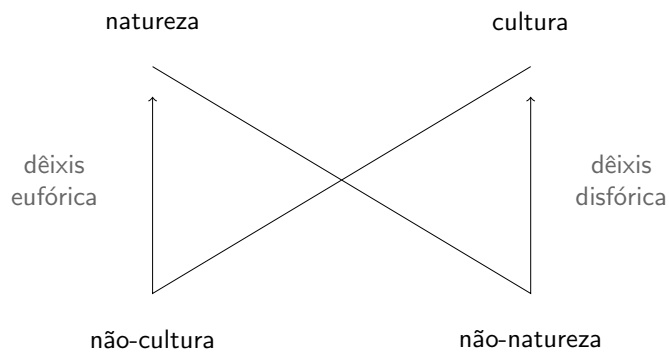


Note no código acima, o uso do comando `\hspace{...}`. Com ele, insere-se um espaço horizontal de extensão determinada pelo usuário. No caso acima, um espaço de 1.3cm será inserido entre a primeira e a segunda árvores.

4.6 Quadrados semióticos e Esquemas tensivos

Para compor os esquemas tensivos e os quadrados semióticos, basta acrescentar no preâmbulo do documento o pacote [esquema-quadrado](#) e inserir o arquivo `esquema-quadrado.sty` no mesmo diretório em que se encontra o arquivo com o texto principal (a saber, o arquivo de extensão `.tex`). O arquivo `esquema-quadrado.sty` contém a programação de duas representações gráficas muito utilizadas em semiótica: (a) o *quadrado semiótico* e (b) o *esquema tensivo*, e por isso não dever ser alterado. Tudo o que você precisa fazer é inserir, no texto principal, os comandos necessários para que o \LaTeX automaticamente busque ou o quadrado semiótico ou o esquema tensivo e atualize-o(s) no arquivo principal (de extensão `.tex`). O único trabalho que você terá é preencher com os devidos termos (por exemplo, “extensidade”, “intensidade”, “natureza”, “cultura” etc.) os campos indicados pelos comandos `\QuadradoSimple`, `\QuadradoComplexo`, `\EsquemaInversoAscendente`, `EsquemaInversoDescendente`, `EsquemaConversoAscendente`, `EsquemaConversoDescendente`.

Vejamos um exemplo de um quadrado semiótico apenas com os termos simples e a indicação das dêixis:



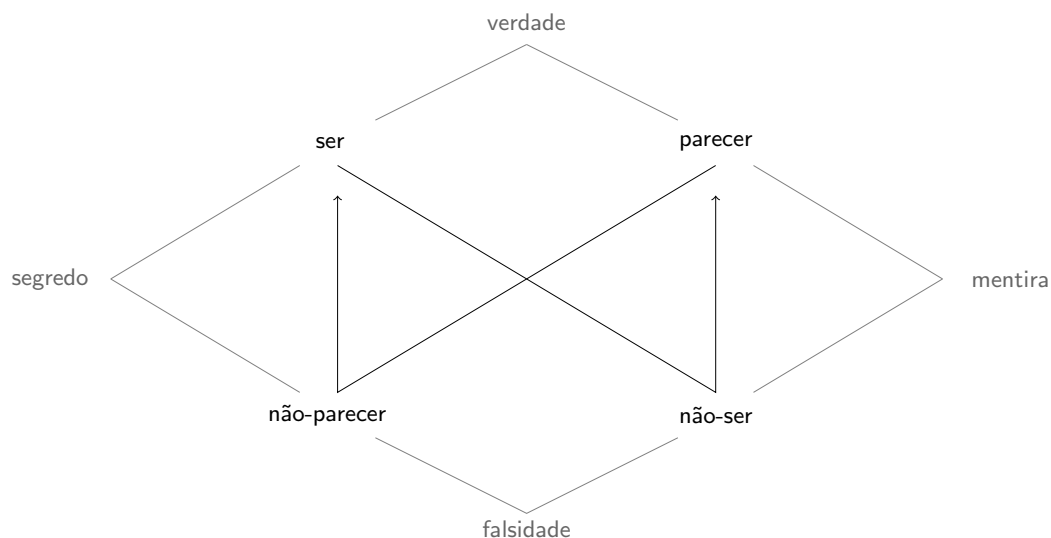
Para compor o quadrado semiótico tal como se apresenta acima, basta acrescentar o comando `\QuadradoSimples` e inserir os termos — como fizemos, com os termos “natureza”, “cultura”, “não-cultura” etc. — nos seis campos disponíveis entre chaves.

```

\begin{center}
\QuadradoSimples
{natureza} %termo superior à esquerda
{cultura} %termo superior à direita
{não-cultura} %termo inferior à esquerda
{não-natureza} %termo inferior à direita
{dêixis\\eufórica} %dêixis à esquerda
{dêixis\\disfórica} %dêixis à direita
\end{center}

```

Vejamos agora um exemplo do quadrado das modalidades veridictórias com todos os termos simples e complexos.



```

\begin{center}
\QuadradoComplexo
{ser} %termo superior à esquerda
{parecer} %termo superior à direita
{não-parecer} %termo inferior à esquerda
{não-ser} %termo inferior à direita
{verdade} %termo complexo
{falsidade} %termo neutro
{segredo} %termo à esquerda
{mentira} %termo à direita
\end{center}

```

Atenção para alguns importantes detalhes: (a) no caso de não serem preenchidos, os campos aparecerão em branco; (b) você pode alterar a formatação das palavras no interior de cada campo preenchido em diferentes parâmetros, como tamanho da fonte (`\small`, `\footnotesize` etc.), família de fonte (`\rm`, `\sf` e `\tt`), negrito (`\bf`), itálico (`\it`), versal versalete (`\sc`); (c) é possível, também, quebrar uma linha, no interior do campo, com o comando das duas barras invertidas `\\` (foi o que fizemos com as duas dêixis, no quadrado semiótico da página anterior).

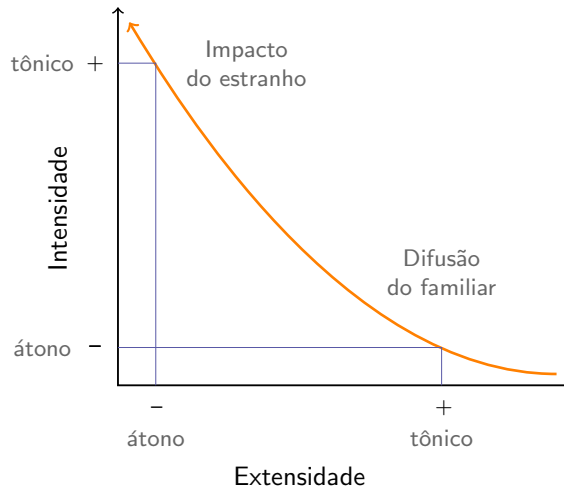
Para compor os esquemas tensivos, o procedimento é semelhante. Basta preencher os campos disponíveis no esquema abaixo:

Esquema tensivo inverso, sentido ascendente:

```

\begin{center}
\EsquemaInversoAscendente
{Intensidade} %eixo da intensid.
{Extensidade} %eixo da extensid.
{tônico} %tônico na intensidade
{átono} %átono na intensidade
{átono} %átono na extensidade
{tônico} %tônico na extensidade
{Impacto\\do estranho}
%mais intenso menos extenso
{Difusão\\do familiar}
%mais extenso menos intenso
\end{center}

```

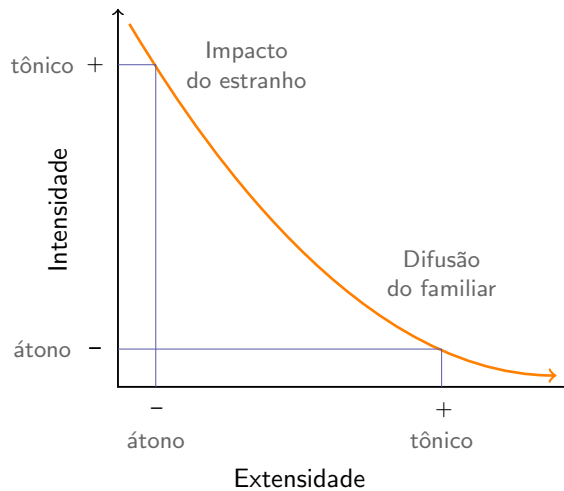


Esquema tensivo inverso, sentido descendente:

```

\begin{center}
\EsquemaInversoDescendente
{Intensidade} %eixo da intensid.
{Extensidade} %eixo da extensid.
{tônico} %tônico na intensidade
{átono} %átono na intensidade
{átono} %átono na extensidade
{tônico} %tônico na extensidade
{Impacto\\do estranho}
%mais intenso menos extenso
{Difusão\\do familiar}
%mais extenso menos intenso
\end{center}

```

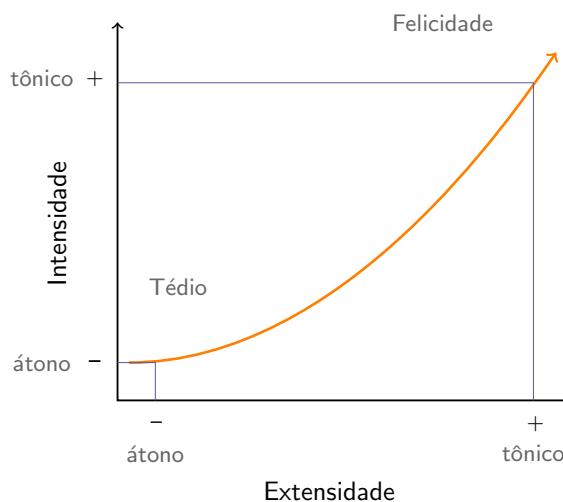


Esquema tensivo converso, sentido ascendente:

```

\begin{center}
\EsquemaConversoAscendente
{Intensidade} %eixo da intensid.
{Extensidade} %eixo da extensid.
{tônico} %tônico na intensidade
{átono} %átono na intensidade
{átono} %átono na extensidade
{tônico} %tônico na extensidade
{Tédio} %menos int. menos ext.
{Felicidade}
%mais intenso mais extenso
\end{center}

```

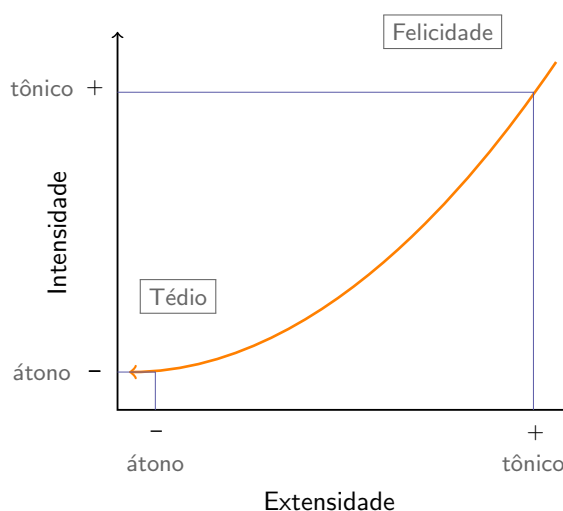


Esquema tensivo converso, sentido descendente:

```

\begin{center}
\EsquemaConversoDescendente
{Intensidade} %eixo da intensid.
{Extensidade} %eixo da extensid.
{tônico} %tônico na intensidade
{átono} %átono na intensidade
{átono} %átono na extensidade
{tônico} %tônico na extensidade
{\fbox{Tédio}}
%menos intenso menos extenso
{\fbox{Felicidade}}
%mais intenso mais extenso
\end{center}

```



Os exemplos apresentados acima foram compostos sobre os *modelos* que desenvolvemos para este curso. Embora tenhamos fixado, em cada uma das representações, o lugar para a inserção dos respectivos termos (“natureza”, “cultura”, “ser”, “parecer”, “Impacto do estranho” etc.), essa posição pode ser alterada, de modo bastante simples, em seu sentido horizontal. Basta, para isso, acrescentar o comando `\hspace` antes do termo entre as chaves, com valores positivos (para mover para direita) ou negativos (para mover para esquerda) da seguinte maneira: `{\hspace{0.8cm}Tédio}` ou `{\hspace{-0.8cm}Tédio}`. No exemplo acima, usamos ainda o comando

`\fbox` para enquadrar os termos “Felicidade” e “Tédio”, destacando-os dos demais.

4.7 Exemplos Numerados

Iremos ilustrar a criação de exemplos numerados e de referências cruzadas relativas a eles através do uso do pacote **linguex**, o qual deve ser declarado no preâmbulo:

```
\usepackage{linguex}
```

Para gerar exemplos numerados, basta utilizar o comando `\ex.` como ilustrado abaixo:

Considere a sentença abaixo:	Considere a sentença abaixo:
<code>\ex.</code> Uma carta chegou.	(1) Uma carta chegou.
Neste caso, pode-se inverter a ordem sujeito-verbo:	Neste caso, pode-se inverter a ordem sujeito-verbo:
<code>\ex.</code> Chegou uma carta.	(2) Chegou uma carta.
Já em outros ...	Já em outros ...

4.7.1 Sub-Exemplos

Para gerar exemplos complexos com mais de um item, utilize os comandos `\a.` e `\b.`

<code>\ex.</code>	(3)	a. Uma carta chegou.
<code>\a.</code> Uma carta chegou.		b. Chegou uma carta.
<code>\b.</code> Chegou uma carta.		

O comando `\a.` indica o primeiro item de um subexemplo. O comando `\b.` é usado do segundo item em diante. Assim se houver mais itens, todos eles serão introduzidos por `\b.` como no caso abaixo:

<code>\ex.</code>	(4)	a. Uma carta chegou.
<code>\a.</code> Uma carta chegou.		b. Chegou uma carta.
<code>\b.</code> Chegou uma carta.		c. Duas cartas chegaram.
<code>\b.</code> Duas cartas chegaram.		d. Chegaram duas cartas.
<code>\b.</code> Chegaram duas cartas.		

Para representar julgamentos de gramaticalidade através dos símbolos usuais (*, ?, % , #), basta inseri-los logo antes do exemplo:

<pre>\ex. \a. Chegaram duas cartas. \b. \% Chegou duas cartas. \b. * Duas cartas chegou.</pre>		(5)	<pre>a. Chegaram duas cartas. b. \%Chegou duas cartas. c. *Duas cartas chegou.</pre>
--	--	-----	--

Se você precisar de um terceiro nível (subsubexemplos), utilize novamente os comandos `\a.` e `\b.` já introduzidos acima.

<pre>\ex. \textbf{Ordem de Palavras} \a. \textit{Espanhol} \a. SVO \b. VOS \b. VSO \z. \b. \textit{Italiano} \a. SVO \b. VOS \b. *VSO</pre>		(6)	<p>Ordem de Palavras</p> <pre>a. <i>Espanhol</i> (i) SVO (ii) VOS (iii) VSO b. <i>Italiano</i> (i) SVO (ii) VOS (iii) *VSO</pre>
---	--	-----	---

Note o uso do comando `\z.` no exemplo acima. Esse comando serve para indicar o término de uma (sub)sub-lista de exemplos, e o retorno para a (sub)lista imediatamente superior. Para encerrar de uma só vez de todos os níveis de uma lista de exemplos, basta deixar uma linha em branco, como havíamos feito nos exemplos anteriores.

4.7.2 Glosas

Para gerar exemplos com glosas, utilize os comandos `\exg.`, `\ag.` e `\bg.` dependendo do nível do exemplo que você está considerando.

<pre>\exg. Can you help me?\ pode você ajudar me\ ‘‘Você pode me ajudar?’’</pre>		(7)	<pre>Can you help me? pode você ajudar me ‘‘Você pode me ajudar?’’</pre>
<pre>\ex. \ag. Can you help me?\ pode você ajudar me\ ‘‘Você pode me ajudar?’’ \bg. Yes, I can.\ sim eu posso\ ‘‘Sim, posso’’</pre>		(8)	<pre>a. Can you help me? pode você ajudar me ‘‘Você pode me ajudar?’’ b. Yes, I can. sim eu posso ‘‘Sim, posso’’</pre>

Note que as duas primeiras linhas de um exemplo com glosas devem terminar com o comando `\` obrigatoriamente.

4.7.3 Referências Cruzadas

Referências cruzadas são obtidas através dos comandos `\label{}` e `\ref{}`. O comando `\label` deve aparecer dentro do exemplo que você deseja mencionar. Ele recebe como argumento um rótulo de identificação, preferencialmente um bom mnemônico, cuja escolha fica a seu critério. Já o comando `\ref` será usado no local em que você deseja fazer referência ao exemplo rotulado, como ilustrado abaixo.

```
\ex.\label{inv}
Chegou uma carta.
```

Como se vê em `\ref{inv}`, é possível inverter sujeito e verbo.

(9) Chegou uma carta.

Como se vê em (9), é possível inverter sujeito e verbo.

Por fim, é importante notar que não há recuo no parágrafo que se inicia logo após um exemplo. Caso você queira tal recuo, deixe após o exemplo duas linhas em branco ao invés de uma.

```
\ex. Uma carta chegou.
```

Nessa linha, não há recuo

(10) Uma carta chegou.

Nessa linha, não há recuo

```
\ex. Uma carta chegou.
```

Nessa linha, há recuo

(11) Uma carta chegou.

Nessa linha, há recuo

Capítulo 5

Apresentações Acadêmicas

Exploraremos neste capítulo a geração de slides e pôsteres para apresentação em conferências acadêmicas

5.1 Slides com *Beamer*

5.1.1 Recursos Básicos

Ilustraremos aqui a criação de slides através do uso do pacote de classe **beamer**. Este pacote gera slides em formato pdf, o que é vantajoso, já que para sua visualização, basta um simples leitor gratuito de pdf (Acrobat Reader, por exemplo). O processamento dos documentos se dá da maneira usual, em que um arquivo com extensão **.pdf** é gerado a partir de um arquivo fonte com a extensão **.tex**.

Há um manual extenso e muito bem elaborado para essa classe, o qual recomendamos fortemente. Nosso objetivo aqui será apenas apresentar os recursos básicos da classe. Entretanto, deve-se notar que mesmo com esses recursos mínimos já é possível montar uma apresentação completa. Para maiores detalhes e recursos mais sofisticados, deve-se consultar o manual.

O primeiro passo é declarar a classe **beamer** na primeira linha de seu documento:

```
\documentclass{beamer}
```

No preâmbulo, declara-se as informações sobre título, autor e data da apresentação:

```
\title{}           %título  
\author{}         %autor(es)  
\institute{}     %afiliação  
\date{}          %data
```

Os conteúdos de todos os slides são especificados no corpo do documento. A forma básica é:

```
\begin{frame}
  <conteúdo do slide>
\end{frame}
```

É bem provável que o primeiro slide de uma apresentação seja o slide em que aparecerão o título, as informações sobre o autor e a data. Essas informações são evocadas pelo comando `\titlepage`:

```
\begin{frame}
  \titlepage
\end{frame}
```

A apresentação propriamente dita começa então a partir do próximo slide. Deste ponto em diante cada slide costuma ter um título:

```
\begin{frame}
\frametitle{0 Que são Morfemas?}
```

Morfemas são as unidades mínimas de uma língua dotadas de significado.

```
\end{frame}
```

É importante salientar que os recursos básicos do Latex estão todos disponíveis para a criação dos slides. Do uso de negrito à confecção de tabelas, tudo isso é compatível com o **beamer**. Além disso, há comandos especiais, como `\alert{}`, que destaca uma porção de texto numa cor especial (vermelho, por exemplo):

```
\begin{frame}
\frametitle{Características Básicas da Língua}
  \begin{itemize}
    \item A ordem canônica é \alert{SOV}.
    \item Apresenta concordância de sujeito e de objeto.
    \item Não há flexão nominal.
    \item Não há artigos.
  \end{itemize}
\end{frame}
```

Um recurso bastante comum em apresentações é mostrar os itens de uma lista um a um, de maneira pausada. Para tanto, existe o comando `\pause`, que é a maneira mais direta de alcançar este efeito:

```

\begin{frame}
\frametitle{Características Básicas da Língua}
\begin{itemize}
\item A ordem canônica é \alert{SOV}.
\pause
\item Apresenta concordância de sujeito e de objeto.
\pause
\item Não há flexão nominal.
\pause
\item Não há artigos.
\end{itemize}
\end{frame}

```

Se você quiser que o item que você está discutindo no momento apareça com uma cor em destaque, você pode utilizar [`<+-|alert@+>`] logo após `\begin{itemize}`. Neste caso, não é necessário usar os comandos `\pause`.

```

\begin{frame}
\frametitle{Características Básicas da Língua}
\begin{itemize}[<+-| alert@+>]
\item A ordem canônica é \alert{SOV}.
\item Apresenta concordância de sujeito e de objeto.
\item Não há flexão nominal.
\item Não há artigos.
\end{itemize}
\end{frame}

```

Uma maneira de se controlar ainda mais os slides em que cada item aparecerá é especificar logo após o comando `\item` o número dos slides correspondentes. Isto é feito da seguinte forma:

```

\begin{frame}
\frametitle{Características Básicas da Língua}
\begin{itemize}
\item<1-> A ordem canônica é \alert{SOV}.
\item<2> Apresenta concordância de sujeito e de objeto.
\item<2-4> Não há flexão nominal.
\item<2,4> Não há artigos.
\end{itemize}
\end{frame}

```

Neste exemplo, o primeiro item aparecerá do primeiro slide em diante, o segundo item aparecerá apenas no segundo slide, o terceiro item do segundo ao quarto slides e o último item apenas no segundo e no quarto slides da seqüência.

De maneira geral, a aparição ou não de qualquer porção de texto pode ser controlada numa seqüência de slides. Para tanto, há dois comandos: `\only<>{}` e `\uncover<>{}`, em que o texto a ser controlado aparece entre chaves. A diferença entre eles é que no caso de **only** o texto selecionado é completamente ignorado em alguns slides e não ocupa qualquer espaço, enquanto que no caso de **uncover**, o texto selecionado ocupa espaço mesmo nos slides em que está encoberto. Teste você mesmo os dois exemplos abaixo e a diferença entre os comandos ficará clara:

```
\begin{frame}
\only<1>{0 menino}\only<2>{A menina} chegou
\end{frame}
```

```
\begin{frame}
\uncover<1>{0 menino}\uncover<2>{A menina} chegou
\end{frame}
```

Considere agora o exemplo abaixo, em que as palavras da primeira sentença vão sendo reveladas (*uncovered*) uma a uma da direita pra esquerda.

```
\begin{frame}
\frametitle{Da Direita pra Esquerda}

\uncover<5->{João} \uncover<4->{comeu} \uncover<3->{a}
\uncover<2->{maça}

\end{frame}
```

Ou ainda exemplo a seguir, em que as palavras vão desaparecendo uma a uma da esquerda pra direita.

```
\begin{frame}
\frametitle{Da Esquerda pra Direta}

\uncover<1>{João} \uncover<1-2>{comeu} \uncover<1-3>{a}
\uncover<1-4>{maça}

\end{frame}
```

Por fim um exemplo mais complexo para você destrinchar. Note o uso do comando `\alert`, que também aceita especificação de slides.

```
\begin{frame}
\frametitle{Hierarquia Sintática}
```

```

\begin{center}

\alert<5->{João} \alert<4->{comeu} \alert<3->{a} \alert<2->{maçã}

\uncover<5->{$_{\text{S}}$} \uncover<5->{$_{\text{DP}}$ João ]}
\uncover<4->{$_{\text{VP}}$} \uncover<4->{$_{\text{V}}$ comeu
]} \uncover<3->{$_{\text{DP}}$} \uncover<3->{$_{\text{D}}$ a ]}
\uncover<2->{$_{\text{NP}}$} maçã ]\uncover<3->{ ]}
\uncover<4->{ ]} \uncover<5->{ ]}

\end{center}
\end{frame}

```

5.1.2 Estruturação

Você pode estruturar suas apresentações com os comandos usuais do Latex, como `\section`, `\subsection`, etc. Dependendo da escolha de formatação (temas) que você fizer (falaremos sobre isso logo em seguida), os títulos dessas seções poderão aparecer no topo ou no rodapé dos slides, mudando automaticamente, à medida que você progride na sua apresentação.

Estruturar sua apresentação permite que o **beamer** crie pra você um sumário (*table of contents*), que pode ser inserido em um slide (geralmente logo após o slide com título/autor/data). Para inserir esse sumário, utilize o comando `\tableofcontents`:

```

\begin{frame}
\frametitle{Roteiro}
\tableofcontents
\end{frame}

```

Se você quiser que as divisões apareçam uma de cada vez, simulando o efeito do comando `\pause`, faça o seguinte:

```

\begin{frame}
\frametitle{Roteiro}
\tableofcontents[pausesections]
\end{frame}

```

5.1.3 Aparência

Há diversos recursos no **beamer** para alterar a aparência dos slides. Aqui vamos nos ater aos chamados **temas**, que são declarados no preâmbulo e que afetam a aparência global da apresentação, incluindo cores, barras de navegação, títulos, listas, etc ...

A opção padrão do **beamer** é por um tema bastante austero, que prima pela simplicidade. Chama-se **default** e, por ser o padrão, não precisa ser declarado explicitamente. Para os demais temas, deve-se declarar o seguinte:

```
\usetheme{<nome do tema>}
```

Não cabe aqui descrever cada um dos temas disponíveis, nem privilegiar alguns deles, já que no fim das contas trata-se mesmo de uma questão de gosto. O que faremos é listar abaixo os nomes de alguns desses temas e sugerir que você teste cada um deles e tire suas próprias conclusões (para a lista completa de opções, consulte o manual).

Alguns temas do beamer: **Boadilla, Madrid, AnnArbor, Juan-LesPins, Montpellier, Berkeley, PaloAlto, Goettingen, Dresden, Frankfurt, Copenhagen, Warsaw**

5.1.4 Handouts

É possível gerar handouts de sua apresentação para serem impressos em papel e eventualmente distribuídos à audiência. Geralmente, opta-se pela colocação de mais de um slide por folha de papel. No exemplo abaixo, são impressos 4 slides por folha.

Em primeiro lugar você deve incluir a opção **handout** à especificação de classe:

```
\documentclass[handout]{beamer}
```

Depois, declare no preâmbulo o uso do pacote **pgfpages** e as especificações mostradas abaixo:

```
\usepackage{pgfpages}
  \pgfpagesuselayout{4 on 1}[a4paper,landscape,border shrink=5mm]
  \mode<handout>{\setbeamercolor{background canvas}{bg=black!5}}
```

Com isso, você irá obter quatro slides por folha com um *background* em cinza bem claro, facilitando assim a visualização.

É bem provável que você queira um tema simples (e geralmente distinto do que você usou na apresentação com slides) na hora de gerar e imprimir o *handout*. Para isso, recomendamos que você suprima a opção de tema que você fez para os slides, o que fará com que os *handouts* sejam produzidos com o tema **default** do **beamer**.

5.2 Posters com *posterusp*

Para a criação de posters a serem apresentados em conferências acadêmicas, recomendamos a utilização de um pacote criado por nós mesmos, denominado

posterusp, e que pode ser baixado [aqui](#). O arquivo **posterusp.sty** deve ser gravado na mesma pasta em que está o arquivo **.tex** que contém o conteúdo do seu poster. O pacote utiliza ainda um arquivo de imagem com o logotipo da USP, que pode ser baixado [aqui](#). Ele também deve estar presente no mesmo diretório do seu texto.

O pacote **posterusp** produzirá posters em **.pdf** prontos para serem impressos em papel tamanho A0 (120cm de altura X 80cm de largura, aproximadamente).

O poster contém um cabeçalho (onde se insere título e autor) e diversos blocos dispostos em duas colunas. Esses blocos funcionam como seções e é neles que o conteúdo de sua apresentação é inserido. Como ilustração, confira o exemplo na página seguinte.

Para produzir seu poster, comece declarando a classe **a0poster** com as opções abaixo:

```
\documentclass[a0,portrait]{a0poster}
```

No preâmbulo, declare o uso do pacote **posterusp**:

```
\usepackage{posterusp}
```

Logo após `\begin{document}`, insira o título e o(s) nome(s) do(s) autor(es) através dos comandos abaixo:

```
\begin{document}
\titulo{...}
\nome{...}
```

O conteúdo do poster propriamente dito vem a seguir e é inserido dentro do ambiente **poster**:

```
\begin{poster}
...
\end{poster}
```

Os blocos que estruturam o poster e que correspondem às seções de sua apresentação são gerados pelo comando **bloco**:

```
\bloco{<tit>}{<cont>}
```

`<tit>` corresponde ao texto que servirá de título para o bloco e `<cont>` é o conteúdo do bloco propriamente dito, valendo aí os comandos usuais do latex. Como exemplo, veja abaixo o código que gerou o primeiro bloco do poster ilustrado a seguir:



Faculdade de Filosofia, Letras e Ciências Humanas

LaTeX para Linguistas

Marcelo Ferreira
Marcos Lopes
Francisco Merçon



1 Preliminares

- O que é LaTeX?
- Instalação e Configuração do Software
- Editores de Texto para LaTeX

4 Ferramentas Linguísticas

- Exemplos Numerados
- Alfabeto Fonético
- Árvores Sintáticas
- Gráficos Tensivos
- Símbolos Especiais

2 Recursos Básicos

- A criação e o Processamento de um Documento
- Texto e Comandos
- Inserção de Autor, Título e Data
- Formatação Mínima
- Divisão do Texto
- Geração de um Sumário

5 Apresentações

- Slides
- Posteres

3 Ambientes

- Listas
- Figuras
- Tabelas
- Citações
- Cabeçalhos e Rodapés

6 Bibliografia

- Bancos de Dados Bibliográficos
- Estilos para as Referências
- A Citação no Texto

Produzido usando LaTeX.

```
\bloco{Preliminares}{  
  
\begin{itemize}  
\item O que é \LaTeX ?  
\item Instalação e Configuração do Software  
\item Editores de Texto para \LaTeX  
\end{itemize} }
```

Cuidado para, quando encerrar um bloco, não esquecer de fechar o parêntese que foi aberto logo após o título!

O processamento do poster se faz da maneira usual, obtendo-se como resultado final um arquivo .pdf pronto para ser enviado para impressão em papel tamanho a0.

Capítulo 6

Referências Bibliográficas

6.1 Preliminares

É muito simples trabalhar com referências bibliográficas no \LaTeX . Há diversas formas de se fazer isso, e este texto tem o intuito de apresentar, dentre essas formas, aquela que é a mais poderosa. Trata-se de uma ferramenta chamada \BIBTeX , desenvolvida pelos mesmos criadores do \LaTeX ¹.

O \BIBTeX busca os dados para gerar as referências em um arquivo externo contendo essas informações em um formato estruturado. Trata-se de um arquivo de texto simples (sem formatação), que funciona com um Banco de Dados bibliográficos, cuja extensão é `.bib`. O comando `\bibliography{}` indica o nome desse arquivo. Assim, o comando

```
\bibliography{Significado}
```

indica que o arquivo de dados bibliográficos é `Significado.bib`. Não é preciso acrescentar a extensão (`.bib`) ao nome do arquivo; ele deve estar preferencialmente presente na mesma pasta onde está o seu arquivo de texto. É também possível trabalhar com arquivos presentes em outras pastas. Nesse caso, deve-se indicar o caminho completo até eles. Se o arquivo estiver, digamos, na pasta `C:\Latex\Biblio`, o comando será `\bibliography{C:/Latex/Biblio/Significado}`. Note-se que as barras separadoras de pastas não são barras invertidas.

Além de indicar o arquivo da bibliografia, será preciso escolher um estilo para as referências bibliográficas, isto é, a forma como elas serão apresentadas, incluindo parâmetros como a ordem dos nomes (“Sobrenome, Nome” ou o contrário), itálicos nos títulos, abreviações, etc. Há literalmente milhares de estilos disponíveis, a maioria associada a padrões de publicações científicas específicas (um estilo para a *Science*, outro para a *Nature*, outro para as teses defendidas na Universidade de Harvard, etc., e até mesmo um

¹Você não precisa baixar separadamente o \BIBTeX . Ele já vem incluído em sua distribuição do \LaTeX (MikTeX) e normalmente já deverá estar instalado em seu computador.

padrão seguindo as normas da ABNT). O comando para definição do estilo bibliográfico é `\bibliographystyle{}`. Ele recebe um parâmetro, que é o identificador do estilo. Por exemplo, o comando

```
\bibliographystyle{apalike}
```

aplica o estilo “apalike” (derivado de APA, *American Psychological Association*) às referências bibliográficas.

Ao final deste texto, na seção *Links*, há sites com outros estilos de referências. Se você quiser experimentar outras possibilidades, alguns dos estilos mais usuais entre as humanidades são:

- `chicago`
- `phiaea`
- `authordate1`
- `amsplain`

6.2 Mãos às obras

Um arquivo chamado *Significado.bib* servirá de ilustração. Ele pode ser baixado [aqui](#). Ele fornece dados para um texto de exemplo, *Teste-bibliográfico.tex*, que deve ser baixado [aqui](#). Baixe os dois arquivos para a mesma pasta em seu disco rígido.

Abra o arquivo *Teste-bibliográfico.tex* no TeXnicCenter. É um arquivo muito simples, com algumas citações de livros e artigos. Repare que ele usa um novo pacote, denominado `natbib`, cuja finalidade é adaptar o padrão das citações para o formato autor–data, além de permitir que sejam empregados muitos estilos de referências bibliográficas bastante úteis para as humanidades e as ciências naturais em geral (daí o nome do pacote).

Para gerar as referências bibliográficas corretamente, esse texto deve ser compilado quatro vezes. Só na última compilação as referências bibliográficas serão efetivamente incorporadas ao texto pronto.

No TeXnicCenter, execute os comandos a seguir, na ordem indicada:

1. Clique o botão *Build current file*
2. Clique o menu *Build / Compile / BibTeX*
3. Clique o botão *Build current file*
4. Clique o botão *Build and view current file*

Feito isso, você deve ter um arquivo de texto compilado com as citações no corpo do texto e as referências bibliográficas completas ao final.

A próxima subseção procura esclarecer a razão dessas quatro compilações e descrever o que se passa entre as diferentes etapas. Se você não tem interesse em se aprofundar no assunto, pode pular para a seção seguinte.

É possível configurar o TeXnicCenter para realizar essa série de operações automaticamente, isto é, sem interrupções. As instruções sobre como fazê-lo podem ser encontradas ao final deste texto.

6.2.1 Por que quatro compilações?

1. A primeira compilação é executada pelo \LaTeX em seu texto e não incorpora as referências. Porém, as chaves das citações (isto é, os parâmetros passados ao comando \cite) encontradas pelo compilador são escritas em um arquivo em separado, com a extensão \.aux . As chaves de identificação são dispostas em ordem crescente nesse arquivo, tal como devem constar na bibliografia ao fim do texto pronto. Além disso, são registradas as informações acerca do arquivo de bibliografia (ou seja, o parâmetro do comando $\text{\bibliography{}}$) e do estilo das citações (parâmetro de $\text{\bibliographystyle{}}$).
2. A segunda compilação é a única feita pelo \BIBTeX propriamente dito. Ele lê o arquivo \.aux e gera um arquivo com a extensão \.bbl . Esse arquivo é muitíssimo importante, pois *pode ser diretamente editado a fim de personalizar a bibliografia* antes de incorporá-la ao texto final. Trata-se de um arquivo de texto bruto (sem formatação), contendo comandos para a produção do texto formatado, como itálicos, aspas, etc.
3. A terceira compilação é feita novamente pelo \LaTeX . Nela, o conteúdo do arquivo \.bbl é incorporado e são lidas as referências cruzadas (*crossref*)² eventualmente presentes na bibliografia, que por sua vez são usadas para atualizar o arquivo \.aux .
4. Na última compilação, é a vez de o \LaTeX ler o arquivo \.aux e usar as informações contidas nele para gerar as referências completas.

Se a sua curiosidade for grande o bastante, você vai querer ver o conteúdo desses arquivos (pela ordem: \.aux , \.bbl e mais uma vez \.aux) durante as etapas intermediárias da compilação. Eles ficam gravados na mesma pasta onde está o seu arquivo \.tex e você pode abri-los usando o TeXnicCenter.

²Neste guia introdutório, não incluí instruções acerca das referências cruzadas. Elas são muito úteis, entre outras coisas, para as referências às obras coletivas, como anais de congressos. Mais informações serão encontradas nos links ao final deste texto.

6.3 Criando e editando os Bancos de Dados bibliográficos

Como foi dito anteriormente, os arquivos `.bib` são escritos em texto simples, sem formatação. Assim, podem ser editados em qualquer editor que não acrescente informações de formatação. O próprio TeXnicCenter seria uma boa escolha. Porém, à medida que esses arquivos vão se tornando grandes, com, digamos, centenas ou milhares de referências, torna-se pouco prático editá-los dessa forma. No entanto, existem alguns programas criados especialmente para esse fim. O mais usado dentre eles é o [JabRef](#), que pode tanto organizar Bancos já existentes como criar novos.

Também é possível usar o [Google Acadêmico](#) para gerar novos registros para o seu Banco de Dados. Não é minha intenção entrar nos detalhes desse procedimento — que é, na realidade, bastante simples; é importante mencionar, no entanto, que você deve ajustar as *Preferências do Google Acadêmico* indicando, na Seção “Gerenciador bibliográfico”, a opção “BibTeX”. Salve suas preferências e execute uma consulta sobre um livro ou artigo qualquer. Você verá, juntamente com os resultados da consulta, o link “Importe para o BibTeX”, que gera uma entrada para o Banco de Dados bibliográfico. Clique nele, copie e cole as informações exibidas em seu arquivo `.bib`.

Por ora, o melhor mesmo é praticar com o arquivo de bibliografia que você já tem em mãos (`Significado.bib`). Ele pode ser aproveitado em seus próprios textos. Para saber quais as chaves de identificação das obras lá presentes (a serem passadas como parâmetros para o comando `\cite`), abra o arquivo no TeXnicCenter — ou, melhor ainda, no JabRef. Naturalmente, esse arquivo pode ser atualizado e ampliado de acordo com suas necessidades.

6.3.1 Organização do arquivo bibliográfico

Para cada referência (livro, artigo, etc.) presente no arquivo `.bib`, um conjunto de informações deve estar presente. A estrutura de cada uma dessas referências é bastante simples, consistindo de uma chave identificadora da obra e de campos de informações típicas, como “Autor”, “Título”, etc. Eis aqui um exemplo:

```
@BOOK{Chomsky72,
  author = {Chomsky, Noam},
  title = {Linguística cartesiana},
  publisher = {São Paulo: Vozes / EDUSP},
  year = {1972}
}
```

A primeira linha é muito importante. Ela apresenta o tipo de obra (que, no caso, é um livro, identificado por `@BOOK`) e contém a chave identificadora

desta referência. Tipicamente, esta segue o padrão *Sobrenome do autor + Ano de publicação (ou de criação)*: daí, no exemplo, “Chomsky72”. Se houver mais de uma obra de um determinado autor num mesmo ano, são acrescentadas letras ao ano: 72a, 72b, 72c. . . Não use acentos nem espaços em branco na criação das chaves.

Deve ser notado também que as informações que preenchem os campos vêm entre chaves, e que os campos são separados entre si por vírgulas. Não há ordem particular de inserção dos campos, com exceção da primeira linha, que corresponde à chave de identificação. Os outros campos (autor, título, etc.) podem ser distribuídos em qualquer ordem, pois serão organizados automaticamente durante a compilação. De forma análoga, as obras podem aparecer em qualquer ordem no arquivo de bibliografia, não sendo necessário organizá-las em função do autor, do ano ou qualquer outro critério.

O preenchimento de alguns campos é opcional e o de outros é obrigatório. Isso depende do tipo da obra citada: artigo, livro, etc. Campos como autor, título e ano são geralmente obrigatórios. Um campo opcional útil é *Note*, muito usado para oferecer indicações sobre o tradutor, por exemplo. Essas informações serão impressas quando forem geradas as referências bibliográficas ao final do texto compilado. Quem escreve uma tese em humanidades tem interesse em mencionar o máximo de informações editoriais possível. O tradutor, o revisor, o autor do prefácio, do posfácio ou das notas críticas e até mesmo o número da edição e a coleção a que eventualmente a obra pertence têm papel importante na concepção do livro.

6.4 Como citar as obras no texto

Quando se usa o pacote `natbib`, as obras são citadas no texto através dos comandos `\citet{}` ou `\citep{}`, conforme se queira ver as citações diretamente no (t)exto ou entre (p)arênteses, respectivamente. Esses comandos recebem como parâmetro obrigatório a chave identificadora da obra e, opcionalmente, o número da página referente à citação. Alguns exemplos:

<code>\citet{Chomsky72}</code>	Chomsky (1972)
<code>\citep{Chomsky72}</code>	(Chomsky, 1972)
<code>\citep[87]{Chomsky72}</code>	(Chomsky, 1972:87)
<code>\citet{chomsky81,chomsky95}</code>	Chomsky (1981,1995)
<code>\citealt{chomsky72}</code>	Chomsky 1972

Ao interpretar o comando `\citet` ou `\citep`, além de inserir a menção à obra no corpo do texto, o `BIBTEX` vai colocá-la no conjunto das Referências Bibliográficas citadas ao final do seu texto.

6.4.1 Citações nos rodapés

Nas ciências humanas, é comum encontrarmos notas bibliográficas nos rodapés. Em geral, essas notas contêm somente as informações editoriais principais (autor, título, editor, ano de edição e números de páginas), deixando de lado alguns detalhes, como nome do tradutor e coleção a que o livro pertence. Além disso, normalmente as referências desse tipo *não* são reproduzidas ao final do texto (como lista de referências). Por fim, uma referência já citada no rodapé costuma ser retomada com *Idem* ou *Op. cit.*, conforme o caso.

O pacote `opcit` permite gerar referências desse tipo. Para usá-lo, declare-o no preâmbulo com o comando `\usepackage{opcit}`.

6.5 Vantagens do uso do BibT_EX

Em contraste com um sistema simples de criação de referências bibliográficas baseado em operações de copiar-e-colar, o BibT_EX apresenta muitas vantagens. Algumas delas são:

- A separação entre o texto e o Banco de Dados bibliográfico permite que você crie um único arquivo com as referências que lhe interessam (ou alguns poucos arquivos divididos tematicamente), digitando uma única vez os dados e podendo usá-los infinitas vezes;
- Seguindo o princípio acima, é possível importar Bancos de Dados já existentes, seja pela Internet, seja através de outros usuários trabalhando em áreas afins;
- As possibilidades de erro na digitação estão presentes uma única vez, precisamente durante a criação do Banco de Dados;
- Não há possibilidade de erros na ordenação das obras citadas na seção de Referências Bibliográficas;
- Você sempre terá a certeza de que uma obra citada no meio do texto comparecerá ao final, na seção de Referências Bibliográficas;
- Inversamente, e igualmente útil, só as obras mencionadas no texto aparecem na seção de Referências Bibliográficas;
- Se for preciso trocar o estilo das citações por exigência de normas editoriais, isso será feito automaticamente, bastando escolher um novo estilo para o comando `\bibliographystyle{}`. Cada vez mais as revistas internacionais têm oferecido seus próprios estilos de formatação para o BibT_EX;
- É possível, inclusive, criar estilos próprios. A maneira mais fácil de fazê-lo é através do pacote `Custom Bib`.

6.6 Opcional: configuração do TeXnicCenter

É possível configurar o TeXnicCenter para realizar automaticamente toda a série de compilações necessárias à produção de um texto completo com as referências bibliográficas ao final. Para isso, siga as instruções a seguir.

1. Clique no menu *Build / Define output profile*
2. Selecione a opção “`latex->pdf`” e clique no botão **Copy**. Na caixa de diálogo que aparecer, apague o nome atual, digite um outro nome à sua escolha (por ex., *GerarBibliografia*) e pressione **Ok**
3. Selecione no quadro à esquerda o perfil que você acabou de criar (*GerarBibliografia*, no nosso exemplo)
4. Clique na aba **PostProcessor**, no quadro à direita, e faça o seguinte:
 - (a) clique duas vezes no quadrado em branco e digite “`pdflatex`”
 - (b) No espaço “`executable`”, digite:³

`C:\Arquivos de Programas\MiKTeX 2.7\miktex\bin\pdflatex.exe`

- (c) No espaço “`Arguments`”, digite o seguinte:

`%pm`

- (d) clique novamente duas vezes no quadrado em branco e repita o processo acima, ou seja, digite `pdflatex` e nos espaços “`executable`” e “`arguments`” digite as mesmas coisas digitadas acima

5. Clique **Ok**

Pronto! Agora basta selecionar na barra de ferramentas do TeXnicCenter a opção que você criou (*GerarBibliografia*, no nosso exemplo). Dessa forma, sempre que você clicar nos botões *Build* com a opção acima selecionada, o compilador rodará o número de vezes necessário para gerar corretamente as referências cruzadas (se houver) e a bibliografia.

6.7 Outros links úteis

- Descrição bastante prática e sucinta do BIB_{TEX} na [Wikipedia](#)

³Naturalmente, esses parâmetros são válidos para a versão 2.7 do MiKTeX e para uma instalação do Windows em português. Caso você possua outra versão do MiKTeX ou do Windows ou, ainda, tenha mudado a pasta de instalação por default do MiKTeX, será preciso ajustar o caminho para o `pdflatex.exe`. Se estiver em dúvida, localize esse arquivo em seu disco rígido e anote o caminho para ele.

- Seção sobre o BIB_EX no breve manual on-line [Getting to Grips with Latex](#)
- [Manual do pacote natbib](#)
- [Tame the BeaST](#), excelente manual cobrindo os aspectos técnicos do BIB_EX
- [Alguns estilos para as bibliografias](#)
- [ABN \$\text{TeX}\$](#) , um conjunto de macros e folhas de estilo para formatar textos e referências bibliográficas nos padrões da ABNT.

Capítulo 7

Seu Texto em L^AT_EX

7.1 O que escrever no preâmbulo dos documentos?

Todo documento no L^AT_EX deve começar baseado no tipo de conteúdo que se pretende diagramar. Quase todo o trabalho de planejamento do documento está representado no preâmbulo, com a escolha da classe do documento e dos pacotes a serem empregados, mas há ainda outras considerações a fazer. O final dos textos acadêmicos também conta com alguns elementos padronizados: bibliografias, índices remissivos, etc. Neste capítulo, veremos os tipos mais comuns de documentos que um autor / pesquisador costuma criar em L^AT_EX.

Os preâmbulos sugeridos aqui contemplam um grande número de pacotes de forma a permitir o emprego de todos os recursos apresentados nesta apostila, além de alguns recursos tipográficos altamente recomendáveis. Entretanto, por essa mesma razão, muitos dos pacotes nos preâmbulos ficarão sem uso na maioria das vezes. Por exemplo, o pacote TIPA não é necessário se você não vai usar os símbolos fonéticos; o pacote Graphicx não é necessário se o documento não usa figuras e assim por diante. Por outro lado, aconselhamos que se evite trocar a *ordem* de declaração dos pacotes que aparecem nos preâmbulos abaixo. Alguns pacotes interagem com outros e não funcionariam bem se carregados em outra posição.

Em todos os casos, os preâmbulos foram concebidos para gerar um arquivo no **formato PDF**. Caso você queira gerar outro formato, será preciso ajustar as opções que mencionam explicitamente o formato PDF no preâmbulo (ex.: `\usepackage[pdftex]{graphicx}`) e excluir a linha referente ao pacote `microtype`.

O maior prejuízo trazido pela presença de pacotes supérfluos no preâmbulo

costuma ser um pequeno atraso no tempo de compilação (sobretudo a primeira compilação). Se isso não for um problema para você, muito bem; caso contrário, você pode simplesmente apagar as declarações dos pacotes que não pretende usar, ou ainda inserir um % no início da linha. Por fim, é evidente que nem todos os pacotes para todos os tipos de preâmbulo estão presentes aqui; e é evidente também que você pode personalizar algumas das opções para mais bem atender às suas preferências.

O mesmo pode ser dito a respeito dos finais dos documentos. Esperamos somente que esses preâmbulos e finais sirvam de pontapé inicial para os seus futuros trabalhos em L^AT_EX.

7.1.1 Artigos

É a classe de documentos mais comum. É usada para se escrever não só artigos como também outros tipos de textos curtos com subdivisões (seções), mas que não tenha de ser dividido em partes ou capítulos (como os livros) nem precise de sumário, índice, etc.

Preâmbulo

```

\documentclass[a4paper,11pt]{article}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{zefonts}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[brazilian]{babel}
\usepackage{verse}
\usepackage{caption}   \captionsetup{justification=centering,labelfont=bf}
\usepackage{color}
\usepackage{endnotes}   \renewcommand{\notesname}{Notas}
\usepackage{epigraph}
\usepackage{fancyhdr}
\usepackage{pdftex}{graphicx}
\usepackage[pdftex, colorlinks=true, urlcolor=blue]{hyperref}
\usepackage{microtype}
\usepackage{multicol}   \setlength\columnseprule{0.4pt}
\usepackage{multirow}
\usepackage{fixltx2e}
\usepackage{natbib}
\usepackage{paralist}
\usepackage{qtree}
\usepackage{stmaryrd}
\usepackage{subfigure}
\usepackage[absolute]{textpos}
\usepackage{tikz}
\usepackage[tone,extra]{tipa}
\usepackage{ulem}
\usepackage{verbatim}
\usepackage{esquema-quadrado}
\usepackage{linguex}   \renewcommand{\refdash}{}

\newcommand{\den}[1]{\$\llbracket\#1\rrbracket\$}

```

```
\title{Meu título}
\author{Eu mesmo}
\date{\today}

\begin{document}

\pagestyle{fancy}
\renewcommand{\sectionmark}[1]{\markboth{#1}{}}
\lhead{}
\chead{}
\rhead{\textit{\small\ \nouppercase{\leftmark}}}
\lfoot{}
\cfoot{\thepage}
\rfoot{}

\setcitestyle{notesep={\,: \,}}

\thispagestyle{plain}
\maketitle

Final

\bibliography{MinhaBibliografia}
\bibliographystyle{apalike} % ou: chicago / phiaea / authordate1 / ...

\end{document}
```


7.1.2 Documentos “oficiais” (pareceres, relatórios, etc.)

Esses são documentos em geral mais curtos e simples do que os artigos e normalmente não contemplam cabeçalhos nem bibliografia ao final. Por outro lado, é comum que haja uma timbre institucional no alto do papel. Ele se encontra aqui representado pelo logotipo da FFLCH (que pode ser baixado [aqui](#) e deve ser colocado no mesmo diretório onde ficará o seu texto) seguido do endereço e telefone do Departamento de Linguística.

Preâmbulo

```
\documentclass[a4paper,12pt]{article}
\usepackage{zefonts}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[brazilian]{babel}
\usepackage{setspace}
\usepackage[usenames]{color}
\usepackage[pdftex]{graphicx}
\usepackage{microtype}
\usepackage[absolute]{textpos}

\begin{document}

\definecolor{cinzento}{gray}{0.5}

\pagestyle{plain}
\thispagestyle{empty}

\begin{textblock*}{297mm}(3mm,13mm)
  \includegraphics[scale=0.4]{LogoFFLCH75.png}
\end{textblock*}

\begin{textblock*}{297mm}(36mm,12mm)

\setstretch{0.7}
\begin{footnotesize}

\sffamily
\color{cinzento}
\noindent
\textbf{Universidade de São Paulo\\
Faculdade de Filosofia, Letras e Ciências Humanas\\
Departamento de Linguística\\}
```

Av. Prof. Luciano Gualberto, 403 --- 01060-970 São Paulo, SP\
Tel: (11) 3091-4298 --- Fax: (11) 3031-6392 --- fll@usp.br

\end{footnotesize}

\end{textblock*}

\setstretch{1}

\color{black}

\vspace{3cm}

\setstretch{1}

\begin{center}

\sffamily

\textbf{PARECER}\

\small

Acerca do pedido de ...

\end{center}

\bigskip

\rmfamily

\normalsize

Final

\bigskip

São Paulo, \today.

\vspace{3cm}

Prof. Dr. ...

\end{document}

7.1.3 Cartas oficiais

As cartas oficiais são semelhantes aos relatórios e pareceres no cabeçalho e diferem destes nos componentes de apresentação (endereços, fórmulas de saudação, etc.).

Preâmbulo

```
\documentclass[a4paper,12pt]{letter}
\usepackage{zefonts}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[brazilian]{babel}
\usepackage{setspace}
\usepackage[usenames]{color}
\usepackage[pdftex]{graphicx}
\usepackage{microtype}
\usepackage[absolute]{textpos}

\signature{Meu Nome}
\address{\textbf{Remetente}}\Endereço do remetente\00000--000 São Paulo, SP}
\date{}

\begin{document}

\definecolor{cinzento}{gray}{0.5}

\pagestyle{plain}
\thispagestyle{empty}

\begin{textblock*}{297mm}(3mm,13mm)
  \includegraphics[scale=0.4]{LogoFFLCH75.png}
\end{textblock*}

\begin{textblock*}{297mm}(36mm,12mm)

\setstretch{0.7}
\begin{footnotesize}

\sfamily
\color{cinzento}
\noindent
\textbf{Universidade de São Paulo}
Faculdade de Filosofia, Letras e Ciências Humanas
```

```

Departamento de Linguística\\}
Av. Prof. Luciano Gualberto, 403 --- 01060-970 São Paulo, SP\\
Tel: (11) 3091-4298 --- Fax: (11) 3031-6392 --- fl1@usp.br
\end{footnotesize}

\end{textblock*}

\setstretch{1}
\color{black}
\rmfamily
\normalsize

\begin{letter}{Il$^{mo}$ S$^{r}$\\\textbf{Destinatário}\\ Setor ou Departamento}

São Paulo, \today.

\opening{Prezado S$^{r}$,}

Final

\closing{Atenciosamente,}

\end{letter}
\end{document}

```

7.1.4 Slides

As diferenças importantes entre os slides e os arquivos é que nos primeiros você deve escolher um padrão temático de cores e outros componentes gráficos. Você pode se contentar com a escolha somente do tema principal, que já traz ótimos ajustes de cores, fontes, cabeçalhos, etc., mas pode também querer ajustar alguns desses parâmetros separadamente. O preâmbulo abaixo começa quase que identicamente ao dos artigos, para que você tenha todos os recursos tipográficos necessários na criação do seu slide, e termina com os parâmetros específicos (e opcionais) para personalizar a aparência do slide, no bloco de linhas imediatamente anteriores à definição do título. Algumas dessas linhas estão precedidas com o comando de comentário (%), pois representam parâmetros raramente modificados pelo usuário.

A seguir, depois de `\begin{document}`, vêm os comandos para a geração de dois slides muito usuais: o slide de apresentação (com título da comunicação, autor, etc.) e um slide resumizando os tópicos a serem apresentados.

Preâmbulo

```

\documentclass{beamer}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{zefonts}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[brazilian]{babel}
\usepackage{verse}
\usepackage{color}
\usepackage{epigraph}
\usepackage{graphicx}
\usepackage{multicol}   \setlength\columnseprule{0.4pt}
\usepackage{multirow}
\usepackage{fixltx2e}
\usepackage{natbib}
\usepackage{paralist}
\usepackage{qtree}
\usepackage{stmaryrd}
\usepackage{subfigure}
\usepackage[absolute]{textpos}
\usepackage{tikz}
\usepackage[tone,extra]{tipa}
\usepackage{ulem}

```

```

\usepackage{verbatim}
\usepackage{esquema-quadrado}
\usepackage{linguex} \renewcommand{\refdash}{}

\newcommand{\den}[1]{\$\llbracket\#1\$\rrbracket\$}

\usetheme{Warsaw}
\usecolortheme[rgb={0.8,0.8,0.8}]{structure}
\mode<presentation>{
\setbeamertemplate{background canvas}
[vertical shading][bottom=yellow!50,top=black!80]
\usefonttheme[onlytiny]{}

% \setbeamertemplate{headline}{} % Elimina o cabeçalho do slide
% \setbeamertemplate{footline}{} % Elimina o rodapé do slide
% \useoutertheme[hooks]{tree} % Tópicos em "árvore" no cabeçalho
% \useinnertheme{rectangles} % Muda a tipografia dos blocos,
%listas numeradas, notas de rodapé e outros

\title{Meu título}
\subtitle{Meu subtítulo}
\institute{DL-USP}
\author{Eu mesmo}
\date{\today}

\begin{document}

\begin{frame}
\titlepage
\end{frame}

\section{Sumário}
\begin{frame}\footnotesize
\frametitle{Sumário}
\tableofcontents[pausesections]
\end{frame}

```

7.1.5 Posters

A fim de facilitar a confecção de posters, criamos o pacote `posterusp`, que automatiza boa parte das tarefas relacionadas. Ele deve ser declarado no preâmbulo dos documentos. O tamanho do papel é importante nos posters. O exemplo a seguir usa o tamanho A0 (841mm de largura por 1189mm de altura), cuja área é de 1m^2 , que é o padrão da maioria dos congressos.

Repare que existe também um ambiente `poster` iniciado após o ambiente `document`.

Preâmbulo

```

\documentclass[a0,portrait]{a0poster}
\usepackage{posterusp}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{zefonts}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[brazilian]{babel}
\usepackage{verse}
\usepackage{color}
\usepackage{epigraph}
\usepackage{graphicx}
\usepackage{multicol} \setlength\columnseprule{0.4pt}
\usepackage{multirow}
\usepackage{fixltx2e}
\usepackage{natbib}
\usepackage{paralist}
\usepackage{qtree}
\usepackage{stmaryrd}
\usepackage{subfigure}
\usepackage[absolute]{textpos}
\usepackage{tikz}
\usepackage[tone,extra]{tipa}
\usepackage{ulem}
\usepackage{verbatim}
\usepackage{esquema-quadrado}
\usepackage{linguex} \renewcommand{\refdash}{}

\newcommand{\den}[1]{\$\llbracket\#1\rrbracket\$}

```

```
\begin{document}

\titulo{Meu Título}
\nome{Meu Nome}

\begin{poster}

Final

\end{poster}

\end{document}
```


7.1.6 Teses e livros

As teses e os livros são classes complexas por seu volume e conteúdo. Além de todos os recursos da classe `article`, as teses e livros incluem *capítulos*, capa, apêndices, sumários e índices remissivos.

Muitas universidades, como Dortmund, Grenoble, Stanford e Oxford, vêm criando classes específicas para seus formatos de tese. Há também classes mais gerais, como a belíssima `ClassicThesis` (e seu pacote complementar `ArsClassica`), destinadas a humanidades, todas facilmente acessíveis através do site [CTAN](#), mas aqui vamos tratar a tese como um livro, usando a classe `book`, por ser esta uma classe bastante simples e abrangente.

Uma das diferenças importantes de um livro para um artigo é que, idealmente, o livro deve ser dividido em *capítulos*. Estes, por sua vez, devem ser arquivos individualizados, de tal forma que seja fácil trabalhar neste ou naquele e igualmente fácil integrá-los depois. Isso se faz através de uma divisão entre os capítulos e um *documento-mestre* que, ao ser compilado, compila todos os arquivos declarados em seu interior, na ordem em que aparecem.

O documento-mestre

O documento-mestre não contém texto propriamente dito, mas somente diretivas para a compilação dos capítulos e outros materiais relacionados. Ele pode ser dividido em duas partes consecutivas, apresentadas abaixo: o preâmbulo e o bloco de inclusão dos conteúdos.

Nosso preâmbulo, como veremos no exemplo de documento-mestre mais adiante, será muito semelhante ao da classe `article`. Logo em seguida (depois de `\begin{document}`), vem a divisão dos conteúdos, representada pelos seguintes segmentos:

1. `\frontmatter`, contendo capa, prefácio, sumário, etc. A numeração das páginas é feita em algarismos romanos
2. `\mainmatter` para os capítulos. A numeração é feita em algarismos arábicos
3. `\backmatter` para a bibliografia
4. `\appendix` para os apêndices, que são facultativos

A cada nova divisão é possível fazer ajustes que serão aplicados a todos os seus segmentos. Isso é útil para gerar um cabeçalho específico para os capítulos diferente daquele da bibliografia, por exemplo.

Divisão em capítulos

Antes de mais nada, para evitar problemas com a compilação, os nomes desses arquivos não devem conter espaços em branco nem os caracteres especiais do L^AT_EX ($\$, _, \dots$), e nem mesmo letras acentuadas.

Os capítulos são incluídos no documento-mestre através do comando `\include{x}`, onde x é o nome (e, opcionalmente, o caminho de diretórios) do arquivo do capítulo. Cada arquivo assim incluído começa em uma página nova, como acontece em um livro (de um bom editor).

A capa

O pacote [titlepages](#) apresenta um grande número de capas para livros e teses com diagramação profissional, em todos os estilos imagináveis, das mais decoradas às mais minimalistas.

Para as teses e dissertações, o código abaixo gera uma capa nos padrões da nossa Faculdade. Não há preâmbulo aqui, pois entende-se que a capa usará o preâmbulo já declarado no início do livro de que ela é parte (isto é, o preâmbulo anterior ao comando `\include{Capa}`).

```

\thispagestyle{empty}
\begin{center}

\small
{\sc Universidade de São Paulo}\
{\sc Faculdade de Filosofia, Letras e Ciências Humanas}\
{\sc Departamento de Linguística}\
{\sc Programa de Pós-Graduação em Semiótica e Linguística Geral}\

\vspace{2cm}

\Large
\mbox{\textbf{Primeira linha do título e}}\
\mbox{\textbf{segunda linha do título}}

\vspace{1cm}

\large
{\sc Fulano De Tal}

\end{center}

```

7.1. O QUE ESCREVER NO PREÂMBULO DOS DOCUMENTOS? 93

```
\vspace{4cm}

\begin{flushright}
\parbox{8cm}{
\small
\noindent
{\sc Dissertação apresentada ao Programa de Pós-Graduação
em Semiótica e Linguística Geral do Departamento de
Linguística da Faculdade de Filosofia, Letras e
Ciências Humanas da Universidade de São Paulo
para a obtenção do título de Mestre em Linguística.}}\}

\bigskip
{\sc Orientador: Prof. Dr. Meu Orientador}
\end{flushright}

\normalsize
\vfill

\begin{center}
São Paulo\\
2010

\end{center}
```

Exemplo de Documento-Mestre

O exemplo a seguir apresenta um preâmbulo, os segmentos de conteúdo e comandos de inclusão de capítulos. Isso corresponde à totalidade de um documento-mestre, com começo, meio e fim.

```
\documentclass[a4paper,12pt,twoside]{book}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{zefonts}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[brazilian]{babel}
\usepackage{verse}
\usepackage{caption}
\captionsetup{justification=centering,labelfont=bf}
```

```

\usepackage{color}
\usepackage{endnotes} \renewcommand{\notesname}{Notas}
\usepackage{epigraph}
\usepackage{fancyhdr}
\usepackage[pdftex]{graphicx}
\usepackage[pdftex, colorlinks=true, urlcolor=blue]{hyperref}
\usepackage{microtype}
\usepackage{multicol} \setlength\columnseprule{0.4pt}
\usepackage{multirow}
\usepackage{fixltx2e}
\usepackage{natbib}
\usepackage{paralist}
\usepackage{qtree}
\usepackage{stmaryrd}
\usepackage{subfigure}
\usepackage[absolute]{textpos}
\usepackage{tikz}
\usepackage[tone,extra]{tipa}
\usepackage{ulem}
\usepackage{verbatim}
\usepackage{esquema-quadrado}
\usepackage{linguex} \renewcommand{\refdash}{}

\newcommand{\den}[1]{\$\llbracket\#1\$\rrbracket\$}

\begin{document}

\frontmatter

    \include{Capa}
    \include{Dedicatoria}
    \include{Agradecimentos}

    \tableofcontents

\mainmatter

    \pagestyle{fancy}
    \renewcommand{\sectionmark}[1]{\markboth{\#1}{}}
    \lhead{}
    \chead{}
    \rhead{\textit{\small\ \nouppercase{\leftmark}}}
    \lfoot{}
    \cfoot{\thepage}

```

7.1. O QUE ESCREVER NO PREÂMBULO DOS DOCUMENTOS? 95

```
\rfoot{}

\setcitestyle{notesep={\,: \,}}

\include{Cap-1}
\include{Cap-2}
\include{Cap-3}
\include{Cap-4}
\include{Cap-5}

\backmatter

\pagestyle{plain}

\addcontentsline{toc}{chapter}{Referências}

\bibliographystyle{chicago}
\bibliography{MinhaBibliografia}

\end{document}
```

7.2 Erros na compilação

Nem sempre a compilação pode ser levada a cabo, por diferentes razões. Ninguém gosta de errar, mas tenha em mente que os erros na produção do seu texto são uma decorrência do seu poder de controle sobre as escolhas do sistema tipográfico. Uma vez detectada e sanada a origem do problema, seu texto será montado exatamente como você gostaria e nunca lhe apresentará surpresas desagradáveis.

7.2.1 Tipos de erros

O L^AT_EX procura sempre interagir com o usuário, anunciando onde e por que a compilação não funcionou. O primeiro passo para sanar o problema é prestar atenção às mensagens exibidas. Elas podem ser vistas no TeXnic-Center na parte inferior da tela, na janela *Output*, e ficam gravadas em um arquivo especial gerado automaticamente durante a compilação chamado *seu-arquivo.LOG*.

Nas mensagens do compilador, é preciso distinguir três tipos de problemas indicados por ele:

Bad boxes Não são propriamente erros, mas avisos acerca de critérios tipográficos. Tudo que o L^AT_EX imprime é diagramado em espaços retangulares delimitados, chamados de *caixas*. Essa mensagem sobre *bad boxes* significa que você talvez tenha preenchido uma das caixas com mais texto do que o sistema esperava. A única consequência associada a isso, quando há, é estética, mas mesmo isso pode ser irrelevante para os seus propósitos, porque o L^AT_EX é muito mais exigente do que o olho humano com relação à diagramação do texto.

Warnings Tampouco são erros, mas advertências sobre potenciais problemas. Em geral, referem-se a referências cruzadas ou referências bibliográficas cujas chaves de referência não foram localizadas durante a compilação. No caso das referências bibliográficas, é normal receber essas mensagens nas três primeiras compilações.

Errors Dos três tipos de mensagens aqui repertoriados, somente este interrompe a compilação, impedindo a produção do arquivo de saída (PDF ou outro). O restante desta seção concentra-se nesse tipo de mensagem.

7.2.2 Erros comuns

A maioria dos erros acontece por simples esquecimento ou desatenção do usuário. Eis aqui alguns deles:

- Esquecer de abrir ou fechar chaves ou colchetes que delimitam argumentos de comandos;
- Esquecer de abrir ou fechar um bloco de ambiente (como, por exemplo, `\begin{document}` sem o respectivo `\end{document}`);
- Trocar o nome de um comando ou esquecer de precedê-lo pela barra invertida;
- Incluir no texto a ser compilado símbolos especiais. O \LaTeX espera compilar um arquivo de texto bruto, ou seja, contendo as letras e números que você vê no teclado. Todos os efeitos gráficos são gerados pelo compilador. Assim, se você copiar e colar um texto do Word, por exemplo, as aspas especiais (“aspas inglesas”) vão ser um problema para a compilação e precisam ser substituídas;
- Mencionar um arquivo (de uma figura, por exemplo) cujo nome não tenha sido digitado corretamente, ou contenha caracteres acetuados, símbolos, etc., ou, ainda, no caso de você usar Linux, que alterne letras maiúsculas e minúsculas (*isto* e *Isto* são coisas diferentes para o Linux);
- Usar recursos de um pacote que não esteja instalado. No Windows, o MikTeX é configurado por default para buscar na internet qualquer novo pacote que seja declarado e instalá-lo automaticamente antes de continuar a compilação do texto. Mesmo assim, esse procedimento pode falhar por conta de dois fatores: sua conexão à internet ou o fato de o pacote requerido não ser registrado na distribuição MikTeX. O primeiro problema é mais comum que o segundo. No caso do segundo, procure na [CTAN](#) as informações de seu pacote insólito;

7.2.3 Como rastrear e eliminar os erros

As mensagens de erro enviadas pelo compilador nem sempre serão descritivas o bastante para que se encontre o problema do texto, pois há mais situações de erro possíveis do que mensagens. Uma boa ideia é preceder as passagens suspeitas com o símbolo de comentário (%) que faz com que elas sejam ignoradas pelo compilador.

Esse procedimento é útil, mas pode ser muito demorado se seu texto for grande. Nesse caso, recomenda-se usar o pacote [verbatim](#). Depois de tê-lo declarado no preâmbulo (`\usepackage{verbatim}`), pode-se isolar um grande bloco de texto com o ambiente `\begin{comment}`/`\end{comment}`. Tudo o que estiver delimitado por esse ambiente será ignorado na compilação. Isolando-se seletivamente grandes porções de texto, é possível chegar rapidamente ao erro.

7.3 Algumas questões frequentes

7.3.1 Generalidades

- O L^AT_EX cria muitos arquivos associados ao texto: além do arquivo TEX (que contém o texto propriamente dito) e o arquivo de visualização (com a extensão PDF, na maioria dos casos), foi criado um arquivo LOG, um arquivo AUX, BBL . . . Posso simplesmente apagá-los?

Sim, desde que você já tenha terminado a sua “sessão de trabalho”, isto é, que você tenha fechado o TeXnicCenter. Esses arquivos servem para organizar sumários, bibliografias e outras informações relativas ao seu arquivo de texto principal (TEX) e são usados nas diferentes etapas da compilação do texto. Quando os arquivos relacionados são excluídos, é preciso gerá-los de novo antes da compilação final. Assim, não seria produtivo apagá-los durante a sua sessão de trabalho, pois isso atrasaria a produção do texto final (do *output*), mas você pode apagá-los sem problemas ao final do dia.

7.3.2 Configuração da página

- Quero definir outras margens para o L^AT_EX. Como fazê-lo?

A configuração de página é feita, no L^AT_EX, de modo bastante simples. A definição das margens laterais e superior/inferior etc. é feita automaticamente, bastando apenas que escolhamos um tamanho para o papel. Assim, ao definir o tamanho `a4paper` ou `letterpaper`, o L^AT_EX estabelece automaticamente a mancha (área do papel onde será impresso o texto), as margens, o cabeçalho, o posicionamento das notas de rodapé e do número de páginas. Note que a configuração padrão do L^AT_EX para todos esses parâmetros é feita com base em conhecimentos tipográficos de alto profissionalismo, levando em conta cálculos do tamanho da fonte, tamanho do papel, espaçamento de linhas, espessura da letra, mancha, margens etc. e a maneira como a percepção humana atua frente a esses parâmetros, com o objetivo de tornar a visualização do texto o mais agradável possível. Por isso é aconselhável valermos-nos, sempre que possível, desses parâmetros predefinidos do L^AT_EX. E por isso, também, deve-se ter sempre cautela ao alterar os parâmetros.

A maneira mais rápida de alterar as margens é através do pacote `fullpage`, de que tratamos no primeiro capítulo. Se ele for insuficiente para suas necessidades e se você quiser ter mais possibilidades ou mais controle sobre o resultado, especificando com valores absolutos o tamanho das margens, use o pacote `geometry`. Será necessário

primeiramente declará-lo no preâmbulo do documento. Repare que todas as opções que vamos utilizar para alterar os parâmetros de margens devem ser inseridas entre colchetes, da seguinte maneira:

```
\usepackage[opção1,opção2,opção3]{geometry}
```

O modo mais simples de se alterar a configuração de página básica do \LaTeX é definindo, como opções, o tamanho do papel, a largura e a altura da mancha. Vejamos o exemplo abaixo:

```
\usepackage[a4paper,textwidth=15cm,textheight=25cm]{geometry}
```

Ao definir, conforme o exemplo acima, o tamanho **A-4** para o papel, **15cm** para a largura da mancha, e **25 cm** para a altura da mancha, o \LaTeX ajustará automaticamente todos os outros parâmetros — a saber, o cabeçalho, os títulos de seções e de subseções, as notas de rodapé etc. — a essa configuração indicada pelo pacote `geometry`, com o intuito de obter o melhor resultado possível.

Uma outra maneira de configurar a página é definindo valores para as margens:

```
\usepackage[a4paper,left=5cm,right=2cm]{geometry}
```

Ou ainda:

```
\usepackage[letterpaper,top=3cm,bottom=3cm]{geometry}
```

Veja que, nos exemplos anteriores, apresentamos primeiro a alteração das margens laterais, e depois das margens superior/inferior. Esse procedimento é importante para que o \LaTeX possa ajustar à sua maneira os outros parâmetros que não sofreram nossa interferência direta.

Vejamos agora um exemplo em que são indicados os parâmetros de todas as quatro margens:

```
\usepackage[letterpaper,top=5cm,bottom=6cm,left=2cm,right=3cm]{geometry}
```

Uma última observação: note que, em todos os exemplos acima, declaramos o tamanho do papel (`a4paper` e `letterpaper`) como opção do pacote `geometry`. Nesse caso, deve-se retirá-lo das opções do `\documentclass`.

7.3.3 Interação do L^AT_EX com outros programas

- Posso converter meu texto feito em Word para o L^AT_EX?

Há alguns programas criados para esse fim. Dentre eles, [word-to-latex](#) oferece uma interface gráfica bastante atraente. O programa só roda nos sistemas operacionais Windows.

Outra alternativa é converter o seu documento no formato DOC para RFT (no próprio Word) e daí fazer a conversão para L^AT_EX. Para tanto foi criado o programa [rtf2latex2e](#), cujas versões são dependentes do sistema operacional: existe uma versão para Mac, uma para Windows (que roda numa linha de comando, através de uma janela DOS) e outra para Linux (que deve ser compilada pelo usuário).

É difícil dizer *a priori* qual dos métodos produz melhor resultado e talvez valha a pena testar dos dois jeitos.

Esses programas podem facilitar um pouco o trabalho, mas não são capazes de converter todo tipo de informação de um arquivo .DOC para um equivalente em L^AT_EX; quanto mais elementos de formatação tiver um determinado objeto, mais distante tende a ficar o resultado da conversão. Além disso, as palavras acentuadas são convertidas para um formato codificado no L^AT_EX (como no ambiente matemático), que prescinde do pacote `inputenc`. Com isso, embora não haja problemas na compilação, a leitura do texto no arquivo TEX fica consideravelmente dificultada.

Por isso, muitos são os que preferem simplesmente copiar e colar o texto do Word em um documento em branco do TeXnicCenter e ir aplicando negritos, quebras de seção, etc., onde necessário. Esse método é sem dúvida o melhor quando o volume do texto a ser convertido é pequeno.

Por fim, se você pretende começar um texto novo, via de regra, é melhor gerá-lo desde o princípio em L^AT_EX.

- Posso converter meu arquivo feito L^AT_EX para o Word?

Pode, mas por que piorar o seu próprio trabalho???

Se a razão para converter um belo texto em L^AT_EX para Word for uma exigência editorial de alguma revista ou evento científico, considere a possibilidade de simplesmente adaptar a configuração de página para atender às exigências do editor. Muitas vezes, aquilo que se exige é um arquivo em PDF com tais e tais margens, esta ou aquela fonte, etc. Se for o caso, você pode mudar essas configurações através dos

comandos relacionados e produzir um resultado quase tão ruim quanto o do Word usando o \LaTeX .

Se o editor exigir um arquivo no formato DOC, há duas saídas. A primeira é simplesmente copiar e colar o texto, apagar o preâmbulo e procurar pelos comandos no texto para também removê-los (a tarefa é facilitada pelo fato de eles sempre estarem associados à barra invertida). A segunda é usar um programa conversor, como [latex2rtf](#), que converte o arquivo TEX para o formato RTF. Assim como no caso inverso, a qualidade do resultado final é dependente da complexidade gráfica do seu texto.

- Posso inserir uma planilha como tabela em um texto para \LaTeX ?

Sim. O modo de fazê-lo é dependente da planilha utilizada. Aconselhamos que se use a planilha **Calc**, do conjunto de aplicativos OpenOffice, pelas seguintes razões: trata-se de um programa livre e gratuito; há versões do programa para todos os sistemas operacionais; é capaz de abrir e manipular planilhas feitas em outros programas, como o Microsoft Excel; seu formato de dados (.ODS) é simples e não tende a mudar com o tempo a fim de gerar obsolescência, como é o caso de alguns programas comerciais; o conjunto de macros de conversão dessas planilhas para o \LaTeX , denominado **Calc2LaTeX**, funciona muito bem. Todas as operações de conversão a seguir devem ser realizadas no programa Calc. Primeiro, será preciso instalar a extensão **Calc2LaTeX**:

1. Clique em *Ferramentas / Gerenciador de Extensão... / Obtenha mais extensões online...*
2. A página de extensões do OpenOffice será aberta em seu navegador. Procure pela caixa de texto de busca nessa página e digite `calc2latex`
3. Nos resultados de sua busca, clique no nome da extensão **Calc2LaTeX**
4. Na página específica da extensão **Calc2LaTeX**, clique no botão *Get it*
5. Baixe a extensão para o seu disco rígido. O nome do arquivo é **Calc2LaTeX**
6. De volta à caixa de diálogo do Gerenciador de Extensão do programa Calc, clique no botão *Adicionar...* e selecione o nome do arquivo baixado.

Isso conclui a instalação. Para converter suas planilhas em tabelas, faça assim:

1. Selecione na planilha ativa as células que deseja converter
2. Clique em *Ferramentas / Macros / Executar macro...*
3. Na lista *Biblioteca*, clique em *Minhas macros / Calc2LaTeX / Calc2LaTeX*. No painel à direita, sob o nome *Macros*, selecione *Main* e clique o botão *Executar*
4. Verifique e ajuste, se for o caso, os parâmetros da macro na caixa de diálogo *Settings*. A seguir, clique no botão *convert*
5. Copie o código que surgirá na caixa de diálogo e cole-o na posição desejada de seu texto no TeXnicCenter.

7.3.4 Outros recursos de uso acadêmico

- Como posso criar gráficos de estatísticas descritivas?

Existem muitas formas, mas o mais aconselhável é usar um pacote estatístico profissional e exportar os gráficos como figuras que serão incluídas no texto da forma habitual (com `\includegraphics{}`). Por sua versatilidade, precisão e plasticidade dos resultados, aconselhamos enfaticamente o software livre [R](#).

- Preciso mandar um resumo para um congresso e eles impõem um limite de n palavras. Como contar as palavras “verdadeiras” de meu texto, sem considerar o preâmbulo, os comandos, etc.?

Experimente usar o programa [TeXcount](#), que roda através de uma interface gráfica via internet.

- Gostaria de produzir as provas para meus alunos em L^AT_EX. Existe algum pacote específico para isso?

Sim. O pacote [Exam](#) permite dividir uma prova em questões com subitens, indicação de pontuação, espaço pautado para respostas e muitos outros recursos.

- É possível gerar certificados ou malas-diretas no L^AT_EX?

Há diferentes pacotes para esse fim. Aquele que recomendamos é o [datatool](#). Os dados deverão estar em um arquivo no formato CSV (*Comma-Separated Values*). O mais prático é usar uma planilha

eletrônica para organizar os dados em campos e a seguir gravá-los (através do menu *Arquivo / Salvar como...*) no formato *Texto CSV*.

7.4 Links (muito) úteis

[LaTeX Cheat Sheet](#). Arquivo em PDF que apresenta em duas páginas os comandos mais comuns em L^AT_EX.

[Usando L^AT_EX](#) Curso completo oferecido pelo Stoa / USP.

[Um manual de L^AT_EX em português](#)

[Seção sobre L^AT_EX dentre os Wikibooks da Wikipedia \(em inglês\)](#)

[Breve descrição dos pacotes disponíveis](#). Eles estendem os recursos originais do L^AT_EX e facilitam a sua utilização. Podem ser instalados diretamente através dos gerenciadores de pacotes no Linux (como o Synaptic) ou por meio do MiKTeX no Windows (*Iniciar/Programas/MiKTeX 2.7/Settings/Packages*).

[The beauty of LaTeX](#). Compara alguns dos recursos tipográficos do L^AT_EX com os processadores de texto "WYSIWYG", como o Word.

7.5 Livros

Se você optar por adquirir livros sobre a produção de documentos com o L^AT_EX, recomendamos fortemente as seguintes obras:

A Guide to L^AT_EX

Fourth Edition

Helmut Kopka e Patrick Daly

Addison Wesley

Harlow, England

2004

Este excelente manual desenvolve em mais profundidade alguns dos tópicos de que tratamos nesta apostila, além de discutir outros que não abordamos. Serve tanto de guia para o iniciante quanto de referência para o iniciado. É extremamente claro e objetivo e contém exercícios práticos que ajudam a fixar o conteúdo apresentado.

The L^AT_EX Companion

Second Edition

Frank Mittelbach e Michel Goossens

Addison Wesley

Boston, MA

2004

Este livro é uma espécie de bíblia do \LaTeX . Em suas mais de mil páginas, você encontrará a descrição detalhada de uma enorme quantidade de pacotes e comandos para os mais diversos fins. Contém um excelente índice remissivo que facilita a busca de informações específicas. É difícil pensar em algum tópico ou problema, por mais peculiar que possa ser, que não esteja incluído nesta obra que impressiona tanto pelo volume quanto pela qualidade.

