

# O MANUAL DO GNU T<sub>E</sub>X<sub>MACS</sub>







# TABELA DE CONTEÚDOS

<b>1. COMEÇANDO</b> . . . . .	9
1.1. Convenções para este manual . . . . .	9
1.2. Configurando o $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ . . . . .	9
1.3. Criando, salvando e abrindo documentos . . . . .	10
1.4. Imprimindo Documentos . . . . .	10
<b>2. ESCRREVENDO DOCUMENTOS SIMPLES</b> . . . . .	11
2.1. Informações gerais sobre digitação de textos . . . . .	11
2.2. Escrevendo texto estruturado . . . . .	11
2.3. Etiquetas de conteúdo . . . . .	12
2.4. Listas . . . . .	12
2.5. Ambientes . . . . .	13
2.6. Diagramação . . . . .	14
2.7. O sistema de seleção de fontes . . . . .	14
2.8. Dominando o teclado . . . . .	15
2.8.1. Regras gerais para prefixos . . . . .	15
2.8.2. Alguns atalhos de teclado padrões . . . . .	15
2.8.3. Atalhos de teclado para o modo texto . . . . .	16
2.8.4. Comandos híbridos e emulação do $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . . . . .	16
2.8.5. Objetos dinâmicos . . . . .	16
2.8.6. Configuração do teclado . . . . .	17
<b>3. FORMULAS MATEMÁTICAS</b> . . . . .	19
3.1. Principais ferramentas matemáticas . . . . .	19
3.2. Digitando símbolos matemáticos . . . . .	20
3.3. Digitando operadores grandes . . . . .	20
3.4. Digitando delimitadores grandes . . . . .	21
3.5. Acentos matemáticos largos . . . . .	22
<b>4. TABELAS</b> . . . . .	23
4.1. Criando Tabelas . . . . .	23
4.2. Modo de formatação . . . . .	23
4.3. Especificando o alinhamento das células e tabelas . . . . .	24
4.4. Especificando o tamanho das células e tabelas . . . . .	24
4.5. Bordas, enchimento e cores de fundo . . . . .	24
4.6. Propriedades avançadas . . . . .	25
<b>5. LIGAÇÕES E CONTEÚDO GERADO AUTOMATICAMENTE</b> . . . . .	27
5.1. Criando rótulos, ligações e referencias . . . . .	27
5.2. Inserindo imagens . . . . .	27
5.3. Gerando o sumário . . . . .	28
5.4. Compilando uma bibliografia . . . . .	28
5.5. Gerando um índice . . . . .	28

5.6. Compilando um glossário . . . . .	29
5.7. Livros e documentos com múltiplos arquivos . . . . .	29
<b>6. FERRAMENTAS DE DIAGRAMAÇÃO AVANÇADA . . . . .</b>	<b>31</b>
6.1. Fluxos . . . . .	31
6.2. Objetos Flutuantes . . . . .	31
6.3. Quebras de página . . . . .	31
<b>7. FERRAMENTAS DE EDIÇÃO . . . . .</b>	<b>33</b>
7.1. Cortar e colar . . . . .	33
7.2. Encontrar e Substituir . . . . .	33
7.3. Verificação ortográfica . . . . .	34
7.4. Desfazendo e refazendo alterações . . . . .	34
<b>8. USANDO O GNU T<sub>E</sub>X<sub>MACS</sub> COMO UMA INTERFACE . . . . .</b>	<b>35</b>
8.1. Uso básico . . . . .	35
8.2. Editando sessões interativas . . . . .	35
8.3. Escolhendo o método de entrada . . . . .	36
<b>9. ARQUIVOS DE ESTILO DO T<sub>E</sub>X<sub>MACS</sub> . . . . .</b>	<b>37</b>
9.1. Arquivos de estilo do T <sub>E</sub> X <sub>MACS</sub> . . . . .	37
9.2. Estilos e pacotes padrão do T <sub>E</sub> X <sub>MACS</sub> . . . . .	37
9.3. Escrevendo seus próprios arquivos de estilo . . . . .	38
9.3.1. Examinando um exemplo . . . . .	38
9.3.2. Diretórios importantes do T <sub>E</sub> X <sub>MACS</sub> . . . . .	38
9.4. Adaptando os arquivos e pacotes padrão do T <sub>E</sub> X <sub>MACS</sub> . . . . .	39
<b>10. THE T<sub>E</sub>X<sub>MACS</sub> PLUG-IN SYSTEM . . . . .</b>	<b>41</b>
10.1. Installing and using a plug-in . . . . .	41
10.2. Writing your own plug-ins . . . . .	41
10.3. Example of a plug-in with SCHEME code . . . . .	43
The <code>world</code> plug-in . . . . .	43
How it works . . . . .	43
10.4. Example of a plug-in with C++ code . . . . .	43
The <code>minimal</code> plug-in . . . . .	43
How it works . . . . .	44
10.5. Summary of the configuration options for plug-ins . . . . .	44
<b>11. RESUMO DAS PRINCIPAIS ETIQUETAS DO T<sub>E</sub>X<sub>MACS</sub> . . . . .</b>	<b>47</b>
11.1. A base comum para a maioria dos estilos . . . . .	47
11.1.1. Marcação Padrão . . . . .	47
11.1.2. Símbolos padrão . . . . .	50
11.1.3. Marcação matemática padrão . . . . .	50
11.1.4. Listas padrão . . . . .	51
11.1.5. Geração de conteúdo automática . . . . .	52
11.1.6. Marcação especial para programas e sessões interativas . . . . .	54
11.2. Ambientes padrão para texto . . . . .	55
11.2.1. Definição de novos ambientes . . . . .	55
11.2.2. Ambiente matemáticos . . . . .	56

---

11.2.3. Ambientes para teoremas e afins . . . . .	56
11.2.4. Ambientes para objetos flutuantes . . . . .	57
11.2.5. Ambientes padrão . . . . .	58
11.3. Headers and footers . . . . .	58
11.4. L <sup>A</sup> T <sub>E</sub> X style sections . . . . .	58
11.5. Markup for automatic numbering . . . . .	59
<b>ÍNDICE</b> . . . . .	<b>61</b>



# CAPÍTULO 1

## COMEÇANDO

### 1.1. CONVENÇÕES PARA ESTE MANUAL

Ao longo de todo o manual do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ , entradas em menus serão tipografadas com uma fonte *sem serifas*, por exemplo: Documento, Arquivo→Carregar or Inserir→Formato da fonte→Itálico. Caracteres que devem ser digitados no teclado serão mostrados com uma fonte *monoespaçada* dentro de caixas, da seguinte forma: `^S`. Você verá atalhos de teclado do lado direito das entradas de um menu, quando estes existirem. As seguintes abreviações são usadas para estes atalhos:

- `↑`. Para combinações precedidas pela tecla shift.
- `^`. Para combinações precedidas pela tecla control.
- `⌘`. Para combinações precedidas pela tecla alt.
- `⌘`. Para combinações precedidas pela tecla meta.
- `⌘⌘`. Para combinações precedidas pela tecla hyper.

Por exemplo, `? B` significa `⌘^B`. Espaços dentro de atalhos de teclado significam que você deve digitar as teclas individualmente. Por exemplo, `⌘T ↑N B` significa `⌘T ↑N B`.

As teclas `⌘`, `⌘` e `⌘⌘` não estão disponíveis em todos os teclados. Em computadores pessoais atuais, a tecla `⌘` em geral é substituída pela tecla `windows`. Quando uma ou mais teclas modificadoras não existem no seu teclado, você pode usar `⌘` ao invés de `⌘`, `⌘` no lugar de `⌘` e `↑F7`, `^` ou `⌘^` no lugar de `⌘⌘`. Por exemplo, `⌘W` equivale a `⌘W`. Você pode também configurar as teclas modificadoras para aproveitar completamente o poderoso conjunto de atalhos de teclado que é fornecido com o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ .

Observe que os menus do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  e o comportamento do teclado são *contextuais*, isto é, dependem do modo atual (“modo texto” ou “modo matemático”), da língua atual e da posição do cursor dentro do texto. Por exemplo, no modo matemático, você tem acesso a atalhos especiais que são úteis para digitar fórmulas matemáticas, mas que são inúteis no modo texto

### 1.2. CONFIGURANDO O $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$

Quando você inicia o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  pela primeira vez, ele automaticamente configura-se da maneira que considera mais apropriada para você. Por exemplo, o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  tentará determinar as configurações de seu sistema para sua língua e para o tipo de papel de sua impressora. No entanto, esta configuração pode algumas vezes falhar, ou você pode preferir usar configuração alternativas. Neste caso, você deve usar o menu Editar→Preferências e escolher suas preferências.

Em particular, recomendamos que você configure a “aparência e comportamento” (“look and feel”) desejado para o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ . O padrão do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  é emular o EMACS, o que fornece alguma compatibilidade entre os atalhos de teclado do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  e do EMACS. Além disso, o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  vem com um sistema poderoso para atalhos de teclado, que tenta otimizar o uso de teclas modificadoras como  $\uparrow$  e  $\wedge$ . Entretanto, em muitos sistemas X Window estas teclas não estão configuradas corretamente, de modo que você pode querer redefinir estes atalhos. Mais detalhes podem ser encontrados na seção sobre a [configuração do  \$\text{T}\_{\text{E}}\text{X}\_{\text{MACS}}\$](#) .

### 1.3. CRIANDO, SALVANDO E ABRINDO DOCUMENTOS

Quando você lança o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  sem nenhuma opção na linha de comando, o editor automaticamente cria um novo documento para você. Você pode também criar um novo documento usando Arquivo→Novo. Documentos recém criados não possuem um nome. Para dar-lhes um, você deve usar Arquivo→Salvar como.

Recomendamos que você dê um nome aos seus documentos imediatamente após sua criação. Isto evita que você perca estes documentos. Também recomendamos que você especifique as propriedades globais de seu documento quando necessário. Primeiramente, você especificar um estilo de documento como artigo, livro ou seminário usando Documento→Estilo. Se você escreve documentos em várias línguas diferentes, você pode escolher a língua do seu documento com Documento→Linguagem. De forma análoga, você pode escolher o tipo de papel usando o menu Documento→Página→Tamanho.

Depois de modificar seu documento, você pode salvá-lo com Arquivo→Salvar. Documentos pré-existentes podem ser abertos com Arquivo→Carregar. Note que você pode editar vários documentos na mesma janela usando o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ ; você pode escolher entre *buffers* diferentes usando `lr` para.

### 1.4. IMPRIMINDO DOCUMENTOS

Você pode imprimir o arquivo atual usando Arquivo→Imprimir→Imprimir tudo. O comportamento padrão do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  é supor que você tenha uma impressora com 600dpi usando papel a4. Estes ajustes podem ser mudados em Preferências→Impressora. Você também pode criar um arquivo postscript usando Arquivo→Imprimir→Imprimir tudo em um arquivo (neste caso, as configuração de impressora são empregadas para a criação do arquivo de saída) ou Arquivo→Exportar→Postscript (neste caso as configurações da impressora são ignoradas).

Quando configurado adequadamente, o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  é garantido ser *wysiwyg*: o resultado da impressão é exatamente o que você vê na sua tela. Para que isto realmente aconteça, você deve, em particular, selecionar Documento→Página→Tipo→Papel e Documento→Página→Layout da tela→Margens como no papel. Você deve certificar-se também que os caracteres na sua tela usam o mesmo número de pontos por polegada que os da sua impressora. Esta precisão na renderização pode ser alterada com Documento→Fonte→Dpi. Atualmente, mudanças tipográficas mínimas podem ocorrer quando se altera o número de pontos por polegada, o que pode alterar o documento globalmente devido a mudanças nas quebras de linhas e páginas. Esta desvantagem pode ser removida em uma versão futura do programa.

## CAPÍTULO 2

### ESCREVENDO DOCUMENTOS SIMPLES

#### 2.1. INFORMAÇÕES GERAIS SOBRE DIGITAÇÃO DE TEXTOS

Assim que você tenha realizado as tarefas iniciais descritas anteriormente, você pode começar a digitar. Os caracteres usuais do Inglês e os sinais de pontuação estão diretamente acessíveis na maioria dos teclados. Caracteres acentuados podem ser obtidos com a tecla de escape. Por exemplo, obtemos “é” digitando `⌘ E`. De maneira análoga, obtemos “à” com `⌘ A` e assim por diante. Palavras longas nos finais de linhas sucessivas são hifenizadas automaticamente. Para hifenizar documentos em línguas estrangeiras corretamente, você deve especificar a língua empregada no documento com o menu Documento→Linguagem.

No lado esquerdo do rodapé da janela principal, você pode ver o estilo do documento e as propriedades do texto na posição atual do cursor. Inicialmente, é mostrado “text roman 10”, que significa que você está no modo texto, usando uma fonte romana com 10 pontos, e sem nenhum estilo de documento escolhido. Você pode mudar as propriedades do texto (fonte, tamanho da fonte, cor, língua) no menu Texto. Você também pode mudar as propriedades do texto que você já digitou selecionando uma região e então alterando estas propriedades usando o menu Texto. Algumas propriedades do texto podem também ser alteradas para todo o documento nos menus Documento→Fonte e Documento→Linguagem.

No lado direito do rodapé é mostrado o caracter ou objeto (por exemplo, uma mudança nas propriedades do texto) exatamente antes do cursor. Também são mostrados todos os ambientes que estão ativos na posição corrente do cursor. Esta informação pode auxiliá-lo a orientar-se no seu documento.

#### 2.2. ESCRIVENDO TEXTO ESTRUTURADO

Normalmente, documentos longos são estruturados: são organizados em capítulos, seções e subseções, contém diferentes tipos de textos, como texto normal, citações, notas de rodapé, teoremas, etc. Depois que você escolhe um *estilo de documento* em Documento→Estilo, o próprio `TEXMACS` cuida da diagramação, gerando automaticamente a numeração de seções, páginas e teoremas, bem como uma diagramação atraente para as notas de rodapé e citações e assim por diante.

Quatro estilos de documentos padrão foram implementados até agora: carta, artigo, livro e seminário. O estilo seminário é usado para criação de transparências. Assim que você tenha escolhido um estilo, você pode organizar seu texto em seções (ver Inserir→Seção), e usar *ambientes* específicos. Exemplos de ambientes são teoremas, proposições, observações e similares (ver Inserir→Ambiente). Outros exemplos são listas de itens (ver Inserir→Listar) ou listas numeradas (ver Inserir→Enumerar).

Quando você se acostumar mais com o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ , será possível adicionar seus próprios ambientes nos seu próprios arquivos de estilo. Suponha, por exemplo, que você frequentemente faz citações e que você deseja que estas apareçam em itálico, com margens de 1cm à esquerda e à direita. Ao invés de mudar manualmente as propriedades do texto e do parágrafo todas as vezes em que você faz uma citação, é melhor criar um ambiente. Não só a criação de uma citação será mais rápida, mas também torna-se possível mudar sistematicamente a diagramação de suas citações ao longo de todo o documento, simplesmente pela mudança da definição deste ambiente. Esta situação pode ocorrer, por exemplo, se você descobre, *a posteriori*, que você prefere que as citações apareçam em uma fonte menor.

### 2.3. ETIQUETAS DE CONTEÚDO

Os exemplos mais simples de estrutura em um texto são as etiquetas de conteúdo. No menu Inserir→content tags você pode ver uma lista destas etiquetas. Etiquetas de conteúdo indicam que uma certa parte do texto é de um tipo particular ou serve a uma finalidade específica. Por exemplo, um trecho importante deve ser marcado com a etiqueta `realçado`. A exibição padrão desta etiqueta usa uma fonte em negrito, como neste **texto realçado**. Texto realçado, no entanto, pode ser exibido de maneira diferente de acordo com o estilo do documento. Por exemplo, texto realçado pode ser renderizado com um cor diferente em transparências para apresentações. Encontra-se a seguir uma pequena lista das etiquetas de conteúdo mais comuns e sua finalidade.

Tag	Example	Purpose
<code>realçado</code>	isto é <b>importante</b>	Indica uma região importante do texto
<code>em</code>	a coisa <i>real</i>	Enfatiza uma região do texto
<code>dfn</code>	Um <i>gnu</i> é um animal	Definição de um conceito
<code>exmp</code>	æ é a ligatura æ	Uma seqüência de caracteres literais
<code>nome</code>	o sistema LINUX	O nome de uma coisa em particular
<code>peessoa</code>	Eu sou JORIS	O nome de uma pessoa
<code>cite*</code>	<i>Moby Dick</i> , de Melville	Uma citação bibliográfica
<code>abrv</code>	Eu trabalho no C.N.R.S.	Uma abreviação
<code>acronimo</code>	o formato HTML	Um acronimo
<code>verbatim</code>	o programa disse <b>hello</b>	Texto literal como a saída de um programa
<code>tecl</code>	Por favor digite <b>return</b>	Texto que deve ser digitado no teclado
<code>code*</code>	cout << 1+1; gera 2	Código fonte de um programa de computador
<code>var</code>	cp <i>src-file dest-file</i>	Variáveis em um programa de computador

Tabela 2.1. Algumas das etiquetas de conteúdo mais comuns.

### 2.4. LISTAS

Usando Inserir→Listar você pode criar uma lista sem numeração. Você pode selecionar uma marca específica como • (bolas), – (travessões) ou → (setas) para indicar entradas na lista, ou usar a marca padrão. Listas pode ser *aninhadas* dentro de outras etiquetas, como na lista abaixo:

- Primeiro item.
- Aqui vem uma sublista:
  - Um sub-item.

- Outro sub-item.
- Um ítem final.

A marca padrão é mostrada de forma diferente dependendo do nível de aninhamento. No nível mais externo, a marca é ●, no segundo nível é ○, e assim por diante. Quando você está dentro de uma lista, note que pressionar ↵ automaticamente começa um novo ítem. Se você necessita de ítems formados por vários parágrafos, então você pode usar ¶ para iniciar um novo parágrafo.

Ambientes de enumeração, que são criados usando Inserir→Enumerar, comportam-se como aqueles criados acima, exceto que os ítems são numerados. Abaixo está um exemplo de enumeração que foi criado com Inserir→Enumerar→Roman:

- I. O primeiro item.
- II. O segundo.
- III. E um último.

O último tipo de listas são listas descritivas. Elas são criadas com Inserir→Descrição e permitem que você descreva uma lista de conceitos:

- Gnu.** Um bicho cabeludo, mas manso.
- Gnat.** Vive apenas nos zoológicos.

## 2.5. AMBIENTES

De forma análoga às etiquetas de conteúdo, ambientes são usados para marcar partes do texto com um significado especial. Entretanto, enquanto [etiquetas de conteúdo](#) normalmente são usadas para marcar pequenos trechos do texto, ambientes freqüentemente estendem-se por vários parágrafos. Alguns ambientes usados comumente em matemática são [teorema](#) e [prova](#), como nos exemplos abaixo:

**TEOREMA 2.1.** *Não existem inteiros positivos  $a$ ,  $b$ ,  $c$  e  $n$  com  $n \geq 3$ , tal que  $a^n + b^n = c^n$ .*

**Demostração.** Não há espaço aqui para escrever a prova. □

Você pode iniciar ambiente usando Inserir→Ambiente. Outros ambientes com exibição similar aos teoremas são [proposição](#), [lema](#), [corolário](#), [axioma](#), [definição](#). Você pode usar o macro [dueto](#) (o macro é ativado digitando `DUETO`) para registrar as pessoas às quais se deve o teorema, como em:

**TEOREMA 2.2.** (PITÁGORAS) *Sob circunstâncias apropriadas, nós temos  $a^2 + b^2 = c^2$ .*

Outros ambientes com exibição semelhante aos teoremas, mas que não enfatizam o texto incluso, são [observação](#), [nota](#), [exemplo](#), [aviso](#), [exercício](#) e [problema](#). Os ambiente remanescentes, [literal](#), [codigo](#), [citar](#), [citação](#) e [verso](#) podem ser usado para digitar texto ou programas com vários parágrafos, citações ou poesia.

## 2.6. DIAGRAMAÇÃO

Como uma regra geral, o próprio  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  é responsável pela diagramação do seu texto. Assim, mesmo sem querer proibir completamente que você o faça, nós não encorajamos que você tente fazer a diagramação do seu documento visualmente. Por exemplo, você não deve inserir espaços ou linhas em branco para criar espaçamento vertical ou horizontal; espaço adicional deve ser inserido explicitamente com `Formato→Espaço`. Isto tornará seu texto mais robusto, no sentido de que você não terá que refazer a diagramação quando realizar pequenas alterações, que podem mudar quebras de linhas ou páginas, ou grandes alterações, como mudar o estilo do documento.

Vários tipos comandos para inserção de espaçamento explícito foram implementados. Você pode inserir espaços rígidos com alturas ou larguras previamente especificadas. Espaços horizontais não tem altura, e são elásticos ou rígidos. O comprimento de um espaço elástico depende da forma com que o parágrafo está sendo hifenizado. Além disto, é possível inserir espaçamento que considera marcas de tabulação. Espaços verticais podem ser inseridos tanto no fim quanto no começo do parágrafo: o espaçamento adicional entre dois parágrafos é o máximo entre o espaçamento vertical antes do segundo parágrafo e o espaço após o primeiro parágrafo (em contraste com o  $\text{T}_{\text{E}}\text{X}$ , isto previne espaçamento desnecessário entre dois teoremas consecutivos).

Quanto à formatação de um parágrafo, o usuário pode especificar o estilo do parágrafo (justificado, alinhado à direita, alinhado à esquerda), as margens do parágrafo e os espaços horizontais à esquerda e à direita no início e no final de cada parágrafo, respectivamente. O usuário também pode controlar o espaçamento entre parágrafos e entre as linhas sucessivas de cada parágrafo.

Você pode especificar a formatação da página no menu `Documento→Página`. Em primeiro lugar, você pode escolher a forma na qual as páginas são mostradas na tela do computador: escolhendo “papel” como o tipo da página em `Documento→Página→Tipo`, você pode ver as quebras de página explicitamente. O padrão é formatar a página como “papiro”, que evita a quebra de páginas durante a preparação do documento. O tipo “automático” admite que o tamanho do papel é exatamente o mesmo da janela do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ . As margens da página e a largura do texto são especificados em `Documento→Página→Layout`. Frequentemente, é conveniente reduzir as margens para exibição na tela do computador; isto é feito em `Documento→Página→Layout` da tela.

## 2.7. O SISTEMA DE SELEÇÃO DE FONTES

No  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ , uma fonte tem 5 características fundamentais:

- Seu nome (roman, pandora, concrete, etc.).
- Sua família (romana, monoespaçada ou sem serifas).
- Seu tamanho (um tamanho base (em pontos) e um tamanho relativo (normal, pequeno, etc.)).
- Sua série (negrito, médio ou leve).
- Sua forma (em pé, itálica, maiúsculas pequenas, etc.).

Perceba que no sistema de seleção de fontes do L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$ , o nome da fonte e sua família são a mesma coisa. Note também que o tamanho base é especificado para todo o documento em Documento→Fonte→Tamanho.

## 2.8. DOMINANDO O TECLADO

### 2.8.1. Regras gerais para prefixos

Já que há tantos atalhos de teclado, é importante ter alguma maneira de classificá-los em várias categorias, para que sua memorização seja mais fácil. Como uma regra geral, atalhos de teclado que são da mesma categoria tem o mesmo prefixo. Os principais prefixos são:

- ^**. Atalhos prefixados pela tecla control são usados freqüentemente para comandos de edição. Eles dependem particularmente do “look and feel” selecionado em Editar→Preferências. Por exemplo, se você usar o look and feel compatível com o EMACS, atalhos da forma **^** correspondem a comandos do EMACS, como **^Y** para colar texto.
- ⌘**. A tecla alt é usada para comandos que dependem do modo no qual você está. Por exemplo, **⌘S** produz texto **realçado** no modo texto e uma raiz quadrada  $\sqrt{\quad}$  no modo matemático. Note que **⌘** equivale a **⌘**.
- ⌘**. A tecla meta é usada para comandos gerais do T<sub>E</sub>X<sub>MACS</sub>, que podem ser usados em qualquer modo. Por exemplo, **⌘!** produz um rótulo. Ela é usada também para outros comandos de edição, como **⌘W** para copiar texto se você usa o look and feel do EMACS. Note que **⌘** equivale a **⌘**.
- ⌘⌘**. A tecla modificadora do usuário é usada para inserir símbolos especiais como letras gregas no modo matemático. Você pode configurar seu teclado para que a tecla **⌘** funcione como a tecla hiper. A tecla **⌘F7** equivale a **⌘⌘**.

Lembramos que as teclas modificadoras usadas para a obtenção dos prefixos **⌘** e **⌘⌘** podem ser *configuradas* em Editar→Preferências.

### 2.8.2. Alguns atalhos de teclado padrões

Alguns atalhos de teclado que são válidos em todos os modos são:

- ⌘↵**. sempre começa um novo parágrafo.
- ⌘⌘**. remove um objeto ou ambiente.
- ⌘\_**. insere um espaço pequeno.
- ⌘?**. insere um espaço pequeno negativo.
- ⌘↶**. posiciona manualmente o início da seleção.
- ⌘↷**. posiciona manualmente o final da seleção.
- ⌘<**. vai para o início do documento.
- ⌘>**. vai para o final do documento.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

`<initial|<collection|<associate|preamble|false)|<associate|language|portuguese|>>>`

### 2.8.3. Atalhos de teclado para o modo texto

Os principais atalhos de teclado no modo texto, são `$` para entrar no modo matemático e os seguintes atalhos padrões para inserir caracteres acentuados:

Shortcut		Example	Shortcut		Example		
<code>⌘' E</code>	Agudo ´	<code>⌘' E</code>	é	<code>⌘` E</code>	Grave `	<code>⌘` E</code>	è
<code>⌘^ E</code>	Circunflexo ^	<code>⌘^ E</code>	ê	<code>⌘^ E</code>	Trema ¨	<code>⌘" E</code>	ë
<code>⌘~ E</code>	Til ~	<code>⌘~ A</code>	ã	<code>⌘↑C</code>	Cedilha ,	<code>⌘↑C C</code>	ç
<code>⌘↑U</code>	Breve ˘	<code>⌘↑U G</code>	ğ	<code>⌘↑V</code>	Check ˇ	<code>⌘↑V S</code>	š
<code>⌘↑O</code>	Anel superior ˆ	<code>⌘↑O A</code>	â	<code>⌘. Z</code>	Ponto superior ˙	<code>⌘. Z</code>	ž
<code>⌘↑H</code>	˘ Húngaro	<code>⌘↑H O</code>	ó				

Tabela 2.2. Digitando caracteres acentuados com o  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ .

Os caracteres especiais  $\text{k}$ ,  $\text{SS}$ ,  $\text{œ}$ ,  $\text{Œ}$ ,  $\text{æ}$  e  $\text{Æ}$  são obtidos digitando-se `⌘F5 S`, `⌘F5 ↑S`, `⌘F5 O E`, `⌘F5 ↑A ↑E`, `⌘F5 A` e `⌘F5 ↑A`, respectivamente.

Em francês, você também pode usar os atalhos especiais `<<` e `>>` para obter os *guillemets* franceses. Em espanhol, os pontos de exclamação e interrogação no início da frase são obtidos com `!*` ou `!`` e `?*` ou `?``, respectivamente.

### 2.8.4. Comandos híbridos e emulação do $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  permite que você entre com comandos do  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  diretamente do teclado. Primeiro você deve digitar a tecla `\` para entrar no modo de comando híbrido  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}/\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ . Em seguida, você deve digitar o comando que você quer executar. Assim que você tenha terminado de digitar o comando, o rodapé esquerdo mostrará algo como:

```
<return>: action to be undertaken
```

Quando você digitar `↵`, seu comando será executado. Por exemplo, no modo matemático você pode criar uma fração digitando `\FRAC↵`.

Se o comando que você digitou não é um comando (reconhecido) do  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , então primeiramente verificamos se o comando é um macro, função ou ambiente (fornecido pelo arquivo de estilo) do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ . Caso isto aconteça, a expansão do macro, a aplicação da função ou a criação do ambiente é feita (com o número correto de argumentos). Caso contrário, supomos que seu comando corresponde a uma variável do ambiente e usamos o seu valor. A tecla `\` é sempre equivalente a um dos comandos `⌘I L`, `⌘I E`, `⌘I A`, `⌘I #` ou `⌘I V`.

### 2.8.5. Objetos dinâmicos

Alguns objetos mais complexos podem ter vários estados durante a edição de um arquivo. Exemplos de *objetos dinâmicos* são rótulos e referências, porque a aparência de uma referência depende de um número determinado dinamicamente. Muitos outros exemplos de marcação dinâmica podem ser encontrados na documentação sobre a [escrita de arquivos de estilo](#).

Na inserção de um objeto dinâmico como um rótulo usando `⌘!`, o estado padrão do objeto é *inativo*. O estado inativo permite que você digite informação que é relevante ao objeto dinâmico, tal como o nome do rótulo, neste caso particular. Alguns objetos dinâmicos podem ter um número arbitrário de parâmetros, e novos parâmetros podem ser inseridos usando-se a tecla `⌘`.

Ao terminar de digitar a informação relevante do objeto dinâmico, você pode digitar `⌘` para *ativar* o objeto. Um objeto dinâmico ativo pode ser desativado colocando o cursor logo após o objeto e digitando `⌘`.

### 2.8.6. Configuração do teclado

O usuário pode modificar o comportamento do teclado. Para fazê-lo, nós sugerimos que você examine os arquivos no diretório `$TEXMACS_PATH/progs/keyboard`, nos quais o comportamento padrão do teclado é definido. Você pode então redefinir o comportamento do teclado no seu arquivo de inicialização particular.



# CAPÍTULO 3

## FORMULAS MATEMÁTICAS

Para digitar fórmulas matemáticas, é necessário que você entre no “modo matemático” digitando a tecla **\$** ou inserindo uma equação (usando Inserir→Matemáticos→Equação). No modo matemático, você tem acesso a comando específicos e combinações de teclas para digitar símbolos matemáticos e fórmulas. Por exemplo, o prefixo **⌘** pode ser usado para inserir símbolos gregos, (lembre-se que **⌘** é equivalente a **F7**, **^** ou **⌘^**).

Este editor privilegia a digitação de fórmulas matemáticas que façam sentido matemático. Esta característica, que será mais desenvolvida em versões posteriores, é útil também para a comunicação com pacotes de álgebra computacional. No presente, você deve, por exemplo, digitar explicitamente o símbolo de multiplicação entre os símbolos  $a$  e  $b$ . Digitar **AB** produz  $ab$  e não  $a b$ .

### 3.1. PRINCIPAIS FERRAMENTAS MATEMÁTICAS

Os principais objetos matemáticos são criados com o prefixo **⌘**, como se segue:

Atalho	Finalidade	Exemplo
<b>⌘\$</b>	Texto	$L = \{x   x \text{ is sufficiently large}\}$
<b>⌘F</b>	Frações	$\frac{a}{b+c}$
<b>⌘S</b>	Raízes quadradas	$\sqrt{x+y}$
<b>⌘⌘S</b>	Raízes $n$ -ésimas	$\sqrt[3]{x^3+y^3}$
<b>⌘N</b>	Negações	$\frac{a}{b/c}$

**Tabela 3.1.** Principal marcação matemática.

Apóstrofes, superscritos e subscritos são criados como mostrado abaixo:

Atalho	Finalidade	Exemplo
<b>'</b>	Primes	$f'$ ou $(g+h)''''$
<b>⌘'</b>	Back-primes	$\sqrt{f}$
<b>_</b>	Subscritos	$x_n$ ou $x_{i_3}$
<b>^</b>	Superscritos	$x^2$ , $x_n^2$ ou $e^{e^x}$
<b>⌘_</b>	Subscritos à esquerda	${}_2x$
<b>⌘^</b>	Superscritos à direita	$\pi_x$ or ${}^*\text{He}^*$

**Tabela 3.2.** Criação de apóstrofes, superscritos e subscritos

### 3.2. DIGITANDO SÍMBOLOS MATEMÁTICOS

Letras gregas são inseridas no  $\text{TEX}_{\text{MACS}}$  através da combinação da tecla modificadora hiper  $\text{⌘}$  com uma letra. Por exemplo, **HyperA** gera  $\alpha$  e **Hyper⌘G** produz  $\Gamma$ . Lembre-se que a tecla  $\text{⌘F7}$  é equivalente a  $\text{⌘}$ , de forma que  $\rho$  pode ser obtido também com **F5 R**. De forma análoga, **F6**, **F7**, **F8** e  $\text{⌘F6}$  podem ser usados para inserir caracteres em negrito, caligráficos, fraktur e blackboard bold. Por exemplo, **F8 M** produz  $\mathfrak{m}$ ,  $\text{⌘F6 ⌘R}$  produz  $\mathbb{R}$  e **F6 F7 ⌘Z** produz  $\mathcal{Z}$ .

Letras gregas também podem ser obtidas como “variações” das letras latinas, através da tecla  $\text{⌘}$ . Por exemplo, **P⌘** gera  $\pi$ . A tecla  $\text{⌘}$  também é usada para obter variações das próprias letras gregas. Por exemplo, tanto **HyperP⌘** quanto **P⌘⌘** produzem  $\varpi$ .

Muitos outros símbolos matemáticos são inseridos com combinações “naturais” de teclas, por exemplo, **->** produz  $\rightarrow$ , **-->** produz  $\longrightarrow$  e **>=** produz  $\geq$ . Analogamente, **|-** produz  $\vdash$ , **|->** produz  $\mapsto$  and **-><-** produz  $\Leftrightarrow$ . Algumas regras gerais são válidas para obter outras variações de símbolos:

- ⌘**. é a principal tecla para obtenção de variações. Por exemplo, **>=** gera  $\geq$ , mas **>=⌘** gera  $\geq$ . Da mesma forma, **<⌘** produz  $\prec$ , **<⌘=** produz  $\preceq$  e **<⌘=⌘** produz  $\preceq$ . Além disso,  $\text{⌘P⌘}$  gera  $\wp$  e **E⌘** gera a constante  $e = \exp(1)$ . Você pode “circular de volta” usando  $\text{⌘⌘}$ .
- @.** é usado para colocar símbolos dentro de caixas, por exemplo **@+** gera  $\oplus$  e **@X** gera  $\otimes$ . Similarmente, **@⌘+** gera  $\boxplus$ .
- /.** é usada para negações. Por exemplo, **=/** gera  $\neq$  and **<=/** gera  $\not\leq$ . Note que **<=⌘/** gera  $\not\leq$ , enquanto **<=⌘⌘/⌘** gera  $\lesseqgtr$ .
- !.** é usado após setas, para forçar com que super e subscritos apareçam sobre e sob as setas. Por exemplo, **-->^X** gera  $\longrightarrow^x$ , mas **-->!^X** gera  $\xrightarrow{x}$ .

Vários outros símbolos que não podem ser inseridos naturalmente como descrito acima, podem ser obtidos usando o prefixo  $\text{⌘F5}$ . A tabela abaixo mostra alguns destes símbolos:

Atalho	Símbolo	Atalho	Símbolo
$\text{⌘F5 A}$	$\amalg$		
$\text{⌘F5 N}$	$\cap$	$\text{⌘F5 U}$	$\cup$
$\text{⌘F5 V}$	$\vee$	$\text{⌘F5 W}$	$\wedge$

**Tabela 3.3.** Alguns símbolos que não podem ser obtidos usando-se as regras gerais.

### 3.3. DIGITANDO OPERADORES GRANDES

As combinações de teclas abaixo são usadas para criar símbolos grandes:

Atalho	Resultado	Atalho	Resultado
<code>↑F5 ↑I</code>	$\int$	<code>↑F5 ↑O</code>	$\oint$
<code>↑F5 ↑P</code>	$\amalg$	<code>↑F5 ↑A</code>	$\amalg$
<code>↑F5 ↑S</code>	$\sum$	<code>↑F5 @+</code>	$\oplus$
<code>↑F5 @X</code>	$\otimes$	<code>↑F5 @.</code>	$\odot$
<code>↑F5 ↑U</code>	$\cup$	<code>↑F5 ↑N</code>	$\cap$
<code>↑F5 ↑V</code>	$\vee$	<code>↑F5 ↑W</code>	$\wedge$

Tabela 3.4. Operadores matemáticos grandes.

O símbolo de integral grande tem duas variações, dependendo de onde você deseja colocar subscritos e subscritos. O comportamento padrão é posicionar os limites como abaixo:

$$\int_0^{\infty} \frac{dx}{1+x^2}.$$

A exibição alternativa, “com limites acima e abaixo”

$$\int \frac{dx}{1+x^2}.$$

é obtida com `↑F5 ↑L ↑I`. Da mesma forma, você pode digitar `↑F5 ↑L ↑O` para obter  $\oint$  com limites.

### 3.4. DIGITANDO DELIMITADORES GRANDES

Delimitadores grandes são criados como mostrado abaixo:

Atalho	Resultado	Atalho	Resultado
<code>math:large (</code>	(	<code>math:large )</code>	)
<code>math:large [</code>	[	<code>math:large ]</code>	]
<code>math:large {</code>	{	<code>math:large }</code>	}
<code>math:large &lt;</code>	<	<code>math:large &gt;</code>	>
<code>math:large /</code>	/	<code>math:large \</code>	\

Tabela 3.5. Atalhos de teclado para delimitadores grandes.

No  $\text{\TeX}_{\text{MACS}}$ , delimitadores grandes podem ser ou “delimitadores à esquerda”, “delimitadores à direita”, ou “delimitadores centrais”. Normalmente, (, [, { e < são delimitadores à esquerda, ), ], } e > são delimitadores à direita, / e \ são delimitadores centrais. Esta atribuição pode ser alterada, no entanto, usando-se as combinações de teclas `math:large L`, `math:large R` e `math:large M`. Por exemplo, `math:large L)` produz ), considerado como um delimitador grande à esquerda.

No  $\text{T}_{\text{E}}\text{X}$  e no  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , “delimitadores centrais”, ou “separadores” não existem; eles são usados para produzir as barras verticais em fórmulas como

$$\left\langle \frac{a}{b+c} \middle| \frac{p}{q+r} \middle| \frac{a}{b+c} \right\rangle.$$

Podem existir tantos delimitadores centrais entre um par de delimitadores à esquerda e à direita quantos forem necessários.

### 3.5. ACENTOS MATEMÁTICOS LARGOS

A tabela abaixo mostra como digitar acentos sobre símbolos ou fórmulas completas. Para a sua conveniência, alguns deles tornam-se tão largos quanto as fórmulas sobrea as quais estão.

Atalho	Exemplo	Varição Larga	Atalho	Resultado
<code>math:large ~</code>	$\tilde{x}$	$\widetilde{x+y}$	<code>math:large ' </code>	$\acute{x}$
<code>math:large ^</code>	$\hat{x}$	$\widehat{x+y}$	<code>math:large `</code>	$\grave{x}$
<code>math:large †B</code>	$\bar{x}$	$\overline{x+y}$	<code>math:large .</code>	$\dot{x}$
<code>math:large †V</code>	$\vec{x}$	$\overrightarrow{AB}$	<code>math:large "</code>	$\ddot{x}$
<code>math:large †C</code>	$\check{x}$	$\overcheck{x+y}$		
<code>math:large †U</code>	$\breve{x}$	$\overbreve{x+y}$		

Tabela 3.6. Atalhos de teclado para acentos matemáticos largos.

# CAPÍTULO 4

## TABELAS

### 4.1. CRIANDO TABELAS

Para criar uma tabela, você pode usar tanto Inserir→Tabela ou um dos atalhos abaixo:

**⌘T ⌘N T**. Criar uma tabela regular.

**⌘T ⌘N ⌘T**. Criar uma tabela regular com células centradas.

**⌘T ⌘N B**. Criar um “bloco”, cujas células são separadas por linhas.

**⌘T ⌘N ⌘B**. Cria um bloco cujas células são centradas.

No modo matemático, algumas outras estruturas similares a tabelas estão disponíveis:

**⌘T ⌘N M**. Criar uma matriz.

**⌘T ⌘N D**. Criar um determinante.

**⌘T ⌘N C**. Criar uma lista de opções.

O ambiente `\eqnarray*` é também um tipo especial de estrutura semelhante a uma tabela, que se estende por toda uma linha. Você pode iniciar uma lista de equações usando Inserir→Matemáticos→Equações.

Em uma tabela recém criada, seu tamanho é o mínimo (em geral  $1 \times 1$ ) e suas células são vazias. Novas linhas e colunas podem ser inseridas com as combinações **⌘←**, **⌘→**, **⌘↑** e **⌘↓**. Por exemplo, **⌘→** cria uma nova coluna à direita da posição corrente do cursor. Você também pode começar uma nova linha após a posição corrente do cursor digitando **↵**.

### 4.2. MODO DE FORMATAÇÃO

No  $\text{TeX}_{\text{MACS}}$ , blocos arbitrários de células podem ser formatados individualmente. Por exemplo, você pode dar a células individuais uma cor de fundo, mas você também pode decidir que uma coluna inteira seja centralizada. Normalmente, comandos de formatação operam sobre células individuais, mas isto pode ser modificado através de Tabela→Modo de operação da célula. Os seguintes modos estão disponíveis:

**⌘T M C**. Operar em células individuais.

**⌘T M H**. Operar em linhas.

**⌘T M V**. Operar em colunas.

**¶T M T**. Operar em toda a tabela.

Também é possível selecionar um bloco de células com o mouse e realizar uma operação sobre este retângulo.

### 4.3. ESPECIFICANDO O ALINHAMENTO DAS CÉLULAS E TABELAS

A operação de formatação mais freqüente é o alinhamento horizontal ou vertical de um bloco de células. Você pode usar as combinações de teclas **¶**, **¶**, **¶** e **¶** para rapidamente alinhar mais para a esquerda, direita, acima ou abaixo.

Um alinhamento específico também pode ser selecionado nos menus Tabela→Alinhamento horizontal da célula e Tabela→Alinhamento vertical da célula. Ou você pode usar atalhos de teclado como **¶T H x** e **¶T V x** para o alinhamento horizontal e vertical, respectivamente.

De forma análoga, você pode especificar como a própria tabela deve ser alinhada com respeito ao texto à sua volta. Isto é feito com o submenus de alinhamento Tabela→Alinhamento horizontal da tabela e Tabela→Alinhamento vertical da tabela, ou usando atalhos de teclado da forma **¶T ↑H x** ou **¶T ↑V x**. Aqui **x** representa **L** para “esquerda”, **C** para “centrado”, **R** for “direita”, **B** for “abaixo” e **T** for “acima”.

### 4.4. ESPECIFICANDO O TAMANHO DAS CÉLULAS E TABELAS

Usando os menus Tabela→Largura da célula→Definir largura e Tabela→Altura da célula→Definir altura você pode especificar a largura e a altura de uma célula. A largura ou altura especificadas podem ser consideradas de três maneiras distintas:

**Modo mínimo.** A dimensão final da célula será o mínimo entre o valor especificado e a dimensão correspondente da caixa dentro da célula.

**Modo exato.** A dimensão da célula será exatamente aquela especificada.

**Modo máximo.** A dimensão final da célula será o máximo entre o valor especificado e a dimensão correspondente da caixa dentro da célula.

A largura da moldura e o enchimento da célula (explicados mais tarde) são consideradas no cálculo do tamanho da caixa dentro da célula.

Você pode também especificar a largura e altura para toda a tabela em Tabela→Propriedades especiais de tabela. Em particular, você pode especificar a largura ou altura da tabela e você escolher como o espaço em branco é distribuído pelas células usando Tabela→Propriedades especiais de célula→Distribuir espaço vazio. A convenção padrão é que o espaço é distribuído igualmente.

### 4.5. BORDAS, ENCHIMENTO E CORES DE FUNDO

Você pode especificar a espessura das bordas e o espaço vazio de enchimento dentro da célula em todas as quatro direções possíveis: à esquerda, direita, acima e abaixo (ver Tabela→Borda da célula). Você tem atalhos de teclado da forma **¶T B x** e **¶T P x** para especificar a espessura da moldura e do enchimento.

A largura padrão das molduras para células no ambiente bloco é `1ln`, isto é, a largura normal da linha na fonte corrente (como a largura de um traço de fração). Esta largura é usada à direita e abaixo de cada célula (exceto quando a célula está na primeira ou última célula). O enchimento horizontal padrão é `1spc`: a largura de um espaço em branco na fonte corrente. O enchimento vertical padrão é `1sep`: a separação padrão mínima entre duas caixas adjacentes.

Cores podem ser atribuídas ao fundo das células com o menu **Tabela**→**Cor de fundo da célula**.

Também é possível atribuir uma moldura e um enchimento padrão para toda a tabela com o menu **Tabela**→**Propriedades especiais de tabela**→**Borda**. Neste caso, o espaço correspondente ao enchimento é aplicado do lado de fora da moldura.

## 4.6. PROPRIEDADES AVANÇADAS

No conjunto de menus, você pode encontrar também muitas outras propriedades interessantes para tabelas. Resumidamente, elas incluem o seguinte:

- Fazer com que uma célula se sobreponha às células vizinhas acima e abaixo.
- Criação de subtabelas inteiras dentro de uma célula.
- Correção da profundidade e da altura do texto, para fazer com que as linhas de base coincidam.
- Hifenização horizontal do conteúdo das células e hifenização vertical da tabela completa.
- Colar várias linhas e/ou colunas juntas, de forma que as células coladas tornam-se “parte da moldura” das células remanescentes.
- Desativação da tabela, para examinar e modificar seu “código fonte”.
- Ajustar o “centro de extensão” da tabela. Depois disto, as propriedades de formatação desta célula serão usadas para novas células criadas em torno deste centro.
- Especificar o tamanho mínimo e máximo da tabela, que será respeitado quando a tabela for modificada (isto é útil principalmente para a criação de macros de tabelas).

Atualmente, todas as tabelas vem dentro de um ambiente como `tabular`, `block`, `matrix`, etc. Quando você criar suas próprias tabelas, você pode usar **Tabela**→**Propriedades especiais de tabela**→**Extrair formato** para extrair o formato de uma dada tabela.



# CAPÍTULO 5

## LIGAÇÕES E CONTEÚDO GERADO AUTOMATICAMENTE

### 5.1. CRIANDO RÓTULOS, LIGAÇÕES E REFERENCIAS

Você pode criar um novo rótulo, que é inicializado desativado, usando `⌘!` ou `Inserir→Link→Etiqueta` e uma referência para este rótulo usando `⌘?` ou `Inserir→Link→Referência`. Tome cuidado e insira o rótulo em um ponto no qual seu número será correto. Quando rotular seções, por exemplo, a posição recomendada é logo após o nome da seção. Quando você rotular equações, o local recomendado é dentro da equação, em seu início.

É possível criar hiper ligações para outros documentos usando `⌘I>` ou `Inserir→Link→Hiperlink`. O primeiro campo da hiper ligação é o texto associado, que é exibido em azul quando a hiper ligação está ativa. O segundo campo contém o nome de um documento, que pode inclusive estar na rede. Como é usual para hiper ligações, uma ligação da forma `#label` aponta para o mesmo documento e uma ligação da forma `url#label` aponta para um rótulo no documento localizado na `url`.

Da mesma forma, uma ação pode ser associada a um trecho de texto ou gráfico usando `⌘I*` ou `Inserir→Link→Ação`. O segundo campo agora contém um script Guile/Scheme, que é executado sempre que você clica duplamente naquele texto, desde que o mesmo esteja ativo. Por motivos de segurança, a execução destes scripts não é automática. O comportamento padrão é perguntar para você se você aceita a execução; isto pode ser alterado em `Opções→Segurança`. Note que o comando Guile/Scheme

```
(system "shell-command")
```

executa `shell-command` como um comando do seu shell.

Finalmente, você pode incluir outros documentos diretamente dentro de um dado documento usando `⌘II` ou `Inserir→Link→Incluir`. Isto permite, por exemplo, que seja incluída uma listagem de um programa no seu texto, de forma que alterações no programa sejam automaticamente refletidas no texto.

### 5.2. INSERINDO IMAGENS

Você pode incluir imagens no texto usando o menu `Inserir→Imagem`. O `TEXMACS` reconhece atualmente os formatos de arquivo `ps`, `eps`, `tif`, `pdf`, `pdm`, `gif`, `ppm`, `xpm` and `fig`. O programa `gs` (`ghostscript`) é usado para exibir imagens postscript. Se o `ghostscript` não estiver instalado no seu sistema, você pode descarregá-lo de

```
www.cs.wisc.edu/~ghost/index.html
```

Os outros tipos de arquivos são convertidos para postscript usando os scripts `tiff2ps`, `pdf2ps`, `pnmtops`, `giftopnm`, `ppmtogif`, e `xpmtoppm`. Se estes scripts não estiverem instalados no seu computador, por favor entre em contato com o seu administrador de sistemas.

O padrão do `TEXMACS` é mostrar as imagens no tamanho em que foram criadas. As operações abaixo funcionam sobre imagens:

- Recortar as imagens com um retângulo de corte. O canto inferior esquerdo da imagem é usado como a origem para o retângulo de recorte.

- Mudar o tamanho de uma imagem. Quando uma nova largura é especificada e altura é deixada livre (ou vice-versa), o tamanho da imagem muda de forma a preservar a razão de aspecto original.
- Aumentar ou reduzir a imagem. Uma maneira alternativa que multiplica a altura e a largura da imagem pela mesma constante

Também incluímos um script para converter figuras, opcionalmente com fórmulas do L<sup>A</sup>T<sub>E</sub>X, em postscript encapsulado. Para incluir uma fórmula do L<sup>A</sup>T<sub>E</sub>X em uma figura do xfig, lembramos que você deveria entrar com a fórmula como texto, selecionando uma fonte do L<sup>A</sup>T<sub>E</sub>X e ativando a opção “special” nas opções de texto.

### 5.3. GERANDO O SUMÁRIO

É muito simples gerar um sumário para o seu documento. Basta colocar o cursor no local onde você deseja a tabela e clicar em Inserir→Automático→Tabela de conteúdos.

Para gerar o sumário, você deve estar em um modo no qual quebras de páginas são visíveis (escolha paper em Documento→Página→Tipo), de modo que as referências aos números de páginas possam ser calculadas corretamente. A seguir, use ou Documento→Atualizar→Tudo para gerar o sumário. Talvez você tenha que repetir isto várias vezes, até que o documento não tenha mais alterações. Afinal, os números das páginas podem mudar devido a mudanças no sumário!

### 5.4. COMPILANDO UMA BIBLIOGRAFIA

Atualmente, o T<sub>E</sub>X<sub>MACS</sub> usa o bibtex para compilar bibliografias. O mecanismo para compilar uma bibliografia automaticamente é o seguinte:

- Escreva um arquivo .bib com todas as suas referências bibliográficas. Este arquivo deve ser formatado como uma bibliografia padrão para o L<sup>A</sup>T<sub>E</sub>X.
- Use Inserir→Link→Citação e Inserir→Link→Citação invisível para inserir citações, que correspondem a entradas no seu arquivo .bib.
- No lugar em a sua bibliografia deve ser compilada, clique em Inserir→Automático→Bibliografia. Você deve responder à pergunta com um estilo do bibtex (como plain, alpha, abbrv, etc.) e seu arquivo .bib.
- Use Documento→Atualizar→Bibliografia para compilar sua bibliografia.

### 5.5. GERANDO UM ÍNDICE

Para criar um índice, você precisa inicialmente adicionar entradas no seu documento usando o menu Inserir→Link→Item do índice. Posteriormente, você deve colocar o cursor no local onde você deseja que o índice seja criado e clicar em Inserir→Automático→Índice. O índice será gerado de forma análoga ao sumário.

No menu **Inserir**→**Link**→**Item** do índice você encontra vários tipos de entradas para índice. As mais simples são “main”, “sub”, “subsub”, que são macros com um, dois e três argumentos, respectivamente. Entradas das formas “sub”, “subsub”, podem ser usadas para subordinar as entradas do índice com respeito a outras entradas.

Uma entrada de índice completa necessita de quatro argumentos. O primeiro é a chave com a qual a entrada será ordenada e ele deve ser uma “tupla” (criada com `#I <`) na qual o primeiro componente é a categoria principal, o segundo uma subcategoria, etc. O segundo argumento de uma entrada complexa pode ser vazio ou então “strong”, indicando que esta entrada aparecerá em negrito no índice. O terceiro argumento normalmente é vazio, porém se você criar duas entradas com o mesmo terceiro argumento não vazio, então isto criará uma “faixa” de números de páginas. O quarto argumento, que é novamente uma “tupla”, é a própria entrada.

Também é possível criar uma linha no índice sem um número de página correspondente, usando “interjeição” em **Inserir**→**Link**→**Item** do índice. O primeiro argumento deste macro é a chave para ordenação da linha no índice e o segundo argumento contém o texto em si. Esta técnica pode ser útil para criar diferentes seções “A”, “B”, etc. no seu documento.

## 5.6. COMPILANDO UM GLOSSÁRIO

Glossários são criados de forma análoga aos índices, mas as entradas não são ordenadas. Um entrada “regular” no glossário apenas contém algum texto e um número de página será gerado para ela. Uma entrada de glossário “descritat” contém um segundo argumento, que explica a notação. Uma entrada “duplicada” também pode ser usada para criar um número de página para a segunda ocorrência de uma entrada. Uma linha de glossário cria um entrada sem um número de página.

## 5.7. LIVROS E DOCUMENTOS COM MÚLTIPLOS ARQUIVOS

Quando um documento ficar muito grande, você pode querer dividi-lo em partes menores. Isto faz com que as partes menores sejam mais facilmente reutilizáveis em outros trabalhos e ainda melhora o desempenho do editor. Um arquivo completo pode ser inserido em outro usando **Inserir**→**Link**→**Incluir**. Para acelerar o processamento dos arquivos incluídos, são guardadas cópias pré-processadas destes arquivos. Para atualizar todos os documentos inclusos, você deve usar **Ferramentas**→**Atualizar**→**Inclusões**.

Ao escrever um livro, normalmente escrevemos cada capítulo em arquivos individuais `c1.tm`, `c2.tm` até `cn.tm`. Em seguida, criamos um arquivo `book.tm` para o livro completo, no qual os arquivos `c1.tm`, `c2.tm` até `cn.tm` são inseridos com o mecanismo descrito acima. O sumário, bibliografia, etc. são, em geral, criadas no arquivo `book.tm`.

Para ver corretamente as referências cruzadas aos outros capítulos durante a edição de um capítulo `ci.tm` em particular, devemos especificar `book.tm` como o “arquivo mestre” para os arquivos `c1.tm` a `cn.tm` usando **Documento**→**Mestre**→**Vincular**. Na implementação corrente, os números dos capítulos não são corrigidos automaticamente com este esquema, e você deve atualizar manualmente a variável do ambiente `chapternr` no início de cada capítulo para obter a numeração correta durante a edição.



# CAPÍTULO 6

## FERRAMENTAS DE DIAGRAMAÇÃO AVANÇADA

### 6.1. FLUXOS

Documentos complexos freqüentemente contêm notas de rodapé ou objetos flutuantes, que são exibidos de forma distinta do corpo principal do texto. Na verdade, o conteúdo de tais documentos complexos é composto de vários *fluxos* independentes, um para o corpo do texto, um para as notas de rodapé, um para os objetos flutuantes e ainda outro para o texto em duas colunas. As quebras de páginas dos diferentes fluxos são feitas de forma muito independente.

Para inserir uma nota de rodapé, você pode usar Formato→Inserção de página→Nota de rodapé. O número de colunas do texto pode ser modificado em Parágrafo→Número de colunas.

### 6.2. OBJETOS FLUTUANTES

Objetos flutuantes podem mover-se na página de modo independente do corpo principal do texto. Normalmente eles contêm figuras ou tabelas que são muito grandes para serem inseridas diretamente no corpo do texto. Um objeto flutuante pode ser inserido com Formato→Inserção de página→Objeto flutuante.

Você também pode criar um objeto flutuante e inserir diretamente a figura ou tabela dentro dele com Formato→Inserção de página→Figura flutuante ou Formato→Inserção de página→Tabela flutuante, respectivamente. Algumas vezes, no entanto, você pode querer inserir várias figuras e tabelas dentro de um único objeto flutuante. Você pode fazer isto com Inserir→Imagem→Figura pequena e Inserir→Tabela→Tabela pequena, respectivamente.

Depois de criar o objeto flutuante, você pode controlar sua posição usando Inserir→Posicionar flutuante (quando o cursor estiver posicionado dentro do objeto). Você pode especificar se quer que o objeto apareça no topo da página, no final da página, ou na página seguinte. O comportamento padrão permite que o objeto apareça em qualquer uma destas posições. Um objeto flutuante, no entanto, nunca será inserido dentro do fluxo de texto normal a menos que existam três ou mais linhas antes do fim ou depois do começo da página.

### 6.3. QUEBRAS DE PÁGINA

As quebras de página podem ser controladas com grande precisão pelo usuário com Documento→Página→Partindo. No submenu Algoritmo, você pode especificar qual o tipo de algoritmo que será usado. Quebra profissional é a que tem o melhor aspecto quando impressa, porém ela pode tornar o programa lento quando usada interativamente, com o modo de papel. O algoritmo de quebra de página ‘relaxado’ é o mais rápido e o algoritmo ‘médio’ é equivalente ao profissional, exceto para material em múltiplas colunas, onde este é muito mais lento.

Você também pode permitir que o algoritmo de quebra de página aumente ou reduza o comprimento das páginas em casos excepcionais usando o submenu Limites. A elasticidade do espaçamento vertical entre parágrafos e afins pode ser especificada no menu Flexibilidade. O fator 1 é o padrão, um fator menor força um espaçamento mais rígido, porém a qualidade das quebras de página pode diminuir.



# CAPÍTULO 7

## FERRAMENTAS DE EDIÇÃO

### 7.1. CORTAR E COLAR

Você pode selecionar texto ou fórmulas clicando e mantendo pressionado o botão esquerdo do mouse. Para apagar a região selecionada, use **Editar**→**Cortar**. Para copiar a seleção, primeiro clique em **Editar**→**Copiar** e depois cole quantas vezes quiser na posição atual do cursor com **Editar**→**Colar**. Você também pode copiar um trecho selecionado com o botão do meio do mouse.

Também é possível mudar as propriedades de uma região selecionada. Por exemplo, para transformar algum texto negro em vermelho, você seleciona a região usando o botão esquerdo do mouse e clica em **Formato**→**Cor**→**Vermelho**. Da mesma forma, se você seleciona uma fórmula e clica em **Inserir**→**Fração**, então a fórmula torna-se o numerador de uma fração

Quando você usa o mecanismo de cortar e colar para trocar dados com outros aplicativos, o texto é copiado e colado usando o formato de dados do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ . Você pode especificar outros formatos para importação e exportação de dados usando os menus **Editar**→**Importar** e **Editar**→**Exportar**, respectivamente. Normalmente, as operações de copiar e colar usam o buffer primário de texto. Usando o menu **Editar**→**Copiar** para você pode especificar tantos outros buffers quanto você quiser.

### 7.2. ENCONTRAR E SUBSTITUIR

Você pode iniciar uma busca pressionando **^S** ou com **Editar**→**Procurar**. Durante a busca, a “cadeia procurada” é mostrada no lado esquerdo do rodapé. Cada caracter que você digita é adicionado a esta cadeia, e a próxima ocorrência da cadeia é mostrada com um quadro vermelho. Quando a tecla **^S** é pressionada uma segunda vez durante uma busca, a próxima ocorrência é procurada. Um sinal sonoro indica que não existem mais outras ocorrências da cadeia procurada; pressionar **^S** novamente faz com que a busca recomece do início do documento. Você pode usar a tecla **⌘** para apagar caracteres digitados durante a busca.

Normalmente, o texto é procurado para a frente, começando da posição corrente do cursos. Você também pode procurar para trás, usando **^R**. Durante a busca, apenas o texto no mesmo modo e na mesma linguagem vai ser examinado. Em outras palavras, quando você busca por  $x$  no modo matemático, você não vai encontrar nenhum  $x$  comum no texto normal. É uma limitação atual do programa que a cadeia de busca só pode conter texto simples e não símbolos matemáticos ou texto estruturado mais complicado.

Uma operação de substituição é iniciada pressionando a tecla **^=** ou **Editar**→**Substituir**. Você tem que informar qual é a cadeia procurada e a cadeia pela qual esta será substituída. A cada ocorrência da cadeia procurada, você deve informar se deseja substituir a cadeia ( $y$ ), não substituí-la ( $n$ ), ou substituir esta e todas as ocorrências posteriores ( $a$ ). Assim como na procura, substituição é limitada ao mesmo modo e à mesma língua.

### 7.3. VERIFICAÇÃO ORTOGRÁFICA

Se o programa `ispell` estiver instalado no seu sistema, então você pode usá-lo para verificar seu texto em relação a palavras digitadas incorretamente. Você pode pressionar `?` ou `Editar→Ortografia`. Note que você pode ter que verificar se os dicionários correspondentes às línguas dos seus textos estão instalados no seu sistema; isto é comum para o Inglês.

Quando você inicia o verificador ortográfico (em todo o texto ou apenas no trecho selecionado do documento), para cada palavra digitada errada você será perguntado sobre qual ação a ser tomada, sendo que no rodapé estão indicadas as opções disponíveis:

- a). Aceite a palavra como está no texto e todas as suas ocorrências futuras.
- r). Substitua a palavra por uma correção que você digitará a seguir.
- i). Indica que palavra está correta e que a mesma deve ser inserida em seu dicionário pessoal.
- 1-9). Escolhe uma das sugestões para substituir a palavra.

Perceba que o `ispell` só corrige palavras digitadas incorretamente. Erros gramaticais não são detectados.

Quando você inicia o verificador ortográfico, ele usará o dicionário da língua ativa no posição atual do cursor (ou no início da região selecionada). Apenas o texto nesta língua será verificado. Se seu documento usa várias línguas diferentes, o verificador deve ser usado para cada uma delas.

### 7.4. DESFAZENDO E REFAZENDO ALTERAÇÕES

É possível desfazer gradualmente todas as mudanças que você fez em um documento a partir do momento em que você iniciou o `TEXMACS`. Isto pode ser feito com o menu `Editar→Desfazer` ou usando as teclas `⌘[` ou `⌘]`. Alterações desfeitas podem ser refeitas com `Editar→Refazer` ou `⌘]`.

Para economizar memória, o número de ações sucessivas que podem ser desfeitas é limitado normalmente a 100. É possível aumentar este número adicionando um comando como

```
(set-maximal-undo-depth 1000)
```

ao seu arquivo pessoal de inicialização (ver `Ajuda→Scheme`). Quando você especifica um número negativo como a profundidade máxima, um número arbitrário de alterações (sujeito à disponibilidade de memória do computador) de alterações podem ser desfeitas.

# CAPÍTULO 8

## USANDO O GNU T<sub>E</sub>X<sub>MACS</sub> COMO UMA INTERFACE

Uma importante característica do T<sub>E</sub>X<sub>MACS</sub> é sua capacidade de se comunicar com sistemas externos, como em sessões de um interpretador de comandos. Tipicamente, é possível executar comandos de um sistema de álgebra computacional externo, e ter os resultados mostrados no modo gráfico. Também é possível executar comandos do interpretador do sistema e programas escritos em SCHEME neste tipo de seção.

### 8.1. USO BÁSICO

Uma seção interativa pode ser iniciada com o menu *Inserir*→*Sessão*. Uma seção consiste de uma seqüência de ambientes de entrada e saída, possivelmente com algum texto entre eles. Quando você pressiona  dentro de um ambiente de entrada de uma sessão, o texto dentro deste ambiente é executado e o resultado é mostrado no ambiente de saída.

Quando você envia um comando para um interpretador, o aplicativo tenta executá-lo. Vários comandos podem ser executados ao mesmo tempo no mesmo documento, mas a saída só será ativa para a sessão na qual o cursor está localizado, e no local exato do cursor. Desta forma, recomendamos que o uso de buffers diferentes para execuções em paralelo.

Para cada tipo de aplicativo externo, você pode escolher entre compartilhar um único processo por todas as sessões abertas, ou iniciar um processo diferente para cada sessão. Mais precisamente, quando você insere uma sessão com *Inserir*→*Sessão*→*Outro*, você pode especificar tanto o “tipo da sessão” (Shell, Pari, Maxima, etc.) quando um “nome da sessão” (o nome padrão é “default”). Sessões com nomes diferentes correspondem a processos diferentes, e sessões com o mesmo nome compartilham um mesmo processo.

Para terminar o processo subjacente a uma dada sessão, você pode usar *Sessão*→*Encerrar sessão*. Quando você tecla  na entrada de um sistema que não está conectado, ele será reiniciado automaticamente. Você também pode usar *Sessão*→*Interromper execução* para interromper a execução de um comando. Vários aplicativos, no entanto, não tem esta funcionalidade.

### 8.2. EDITANDO SESSÕES INTERATIVAS

Dentro dos campos de entrada de sessões interativas, as teclas de cursor tem significado especial: quando você move o cursor para cima ou para baixo, você move a entrada para os campos de entrada anteriores ou posteriores. Quando você usa as teclas de movimento para a esquerda ou direita, você nunca deixa o campo de entrada atual, você precisa usar o mouse para mover o cursor lateralmente para fora de um campo de entrada.

Os menus `Sessão→Insert fields` e `Sessão→Remove fields` fornecem algumas facilidades para editar os campos de entrada, saída e textos. A maioria das operações aplica-se diretamente a um par correspondente de campos de entrada e saída. Opcionalmente, um campo com um texto explicativo pode ser associado com um campo de entrada com `Sessão→Insert fields→Insert text field`. Atalhos do teclado para inserção de campos são `⌘↑` (insere acima) e `⌘↓` (insere abaixo). Atalhos de teclado para campos de texto/entrada/saída correspondentes são `⌘⌘` (remove para trás) e `⌘⌘` (remove os campos correntes).

É possível criar “sub-sessões” usando `Sessão→Insert fields→Fold input field` ou `⌘→`. Neste caso, o campo corrente de texto, entrada ou saída torna-se o corpo de uma sub-sessão “desdobrada”. Esta sub-sessão consiste de um texto explicativo junto a uma seqüência de campos de entrada e saída. Subsessões podem ser “dobradas” e “desdobradas” com `⌘?` e `⌘?`, respectivamente. A formatação gráfica das subsessões na tela é muito atraente quando se usa o pacote `framed-session` que está em `Documento→Usar pacote→Programa`.

Outras operações de edição úteis para campos de text/entrada/saída são `Sessão→Remove fields→Remove all output fields`, que é útil para criar sessões de demonstração que serão executadas mais tarde, e `Sessão→Split session`, que pode ser usada para dividir uma seção em várias partes para inclusão em um artigo.

### 8.3. ESCOLHENDO O MÉTODO DE ENTRADA

O comportamento padrão do T<sub>E</sub>X<sub>MACS</sub> é tentar executar o campo de entrada quando a tecla `↵` é pressionada. Uma entrada com várias linhas pode ser criada usando-se `⌘↵`. Ou, de forma alternativa, você pode escolher `Sessão→Input mode→Multiline input`, e a tecla `↵` comporta-se da maneira normal e a tecla `⌘↵` é usada para executar o campo de entrada. Observe também que alguns sistemas tem heurísticas internas para determinar quando a entrada está completa; caso não esteja, então a tecla `↵` comporta-se da maneira usual.

Alguns aplicativos permitem que você entre com fórmulas matemáticas de maneira gráfica, em duas dimensões. Esta característica pode ser usada escolhendo `Sessão→Input mode→Mathematical input`. Se esta característica está disponível, então geralmente também é possível copiar e colar a saída de volta na entrada. Depende muito de cada aplicação, no entanto, quão bem isto funciona.

([tmdoc-copyright](#)|1998–2003|Joris van der Hoeven|Ramiro Brito Willmersdorf)

# CAPÍTULO 9

## ARQUIVOS DE ESTILO DO $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$

### 9.1. ARQUIVOS DE ESTILO DO $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$

Uma das melhores características do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  é a possibilidade de escrever seus próprios arquivos de estilo. Arquivos de estilo tem múltiplas funções:

- Permitem a abstração de elementos repetitivos em textos como seções, teoremas, enumerações, etc.
- Formam um mecanismo que permite que você estruture o seu texto. Por exemplo, você pode indicar que um trecho do seu texto é uma abreviação, uma citação ou “importante”.
- Alguns estilo padrão permitem que você escreva documentos com aparência profissional, porque foram projetados com muito cuidado por pessoas com muito conhecimento muito sobre tipografia e estética.

É possível associar um ou mais estilos a um documento. O estilo principal do documento é escolhido no menu Documento→Estilo, e estilos adicionais podem ser adicionado a partir de Documento→Usar pacote.

Do ponto de vista do programa, cada estilo corresponde a um arquivo `.ts`. Os arquivos correspondentes a cada estilo são processados como se fossem documentos comuns, porem o editor mantém apenas o ambiente final de cada arquivo de estilo como o ambiente inicial do documento. Os arquivos de estilo são processados na ordem em que foram listados, assim como os arquivos de estilo usados dentro destes, recursivamente.

### 9.2. ESTILOS E PACOTES PADRÃO DO $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$

Estão implementados atualmente os seguintes estilos:

- Book (livro);
- Article (artigo);
- Letter (carta);
- Seminar (seminário, para transparências).

Cada um destes estilos exporta um certo número de funções e ambientes padronizados listados abaixo. Todos os formatos padrão do futuro deverão suprir pelo menos os comandos e ambiente abaixo, e nós sugerimos que usuários que implementem seus próprios arquivos de estilo façam o mesmo.

- Divisão to texto em seções.

- Ambientes para listas numeradas e não numeradas.
- Ambientes semelhantes a equações.
- Ambientes semelhantes a teoremas.
- Ambientes de programação

Chamamos atenção para o fato de que ambientes para teoremas não são padronizados no L<sup>A</sup>T<sub>E</sub>X, e isto é uma grande fonte de incompatibilidades. Novos “teoremas” podem ser adicionados com o comando `newtheorem`. Também é possível introduzir novos “comentários” com o comando `newremark`; “comentários” são distintos de teoremas no sentido em que não são tipografados com uma fonte enfatizada.

Claramente, ambientes de programação também não existem no L<sup>A</sup>T<sub>E</sub>X. Tais ambientes encontram-se em desenvolvimento.

### 9.3. ESCRREVENDO SEUS PRÓPRIOS ARQUIVOS DE ESTILO

Quando os arquivos de estilo padrão do T<sub>E</sub>X<sub>MACS</sub> não são adequados para uma dada finalidade, você pode escrever seus próprios arquivos. Começar do zero, no entanto, é uma tarefa muito complexa. Assim, recomendamos a reutilização e modificação dos arquivos e pacotes padrão sempre que possível. Consequentemente, é recomendável que primeiramente você leia um pouco sobre a [adaptação](#) de arquivos de estilo e pacotes do T<sub>E</sub>X<sub>MACS</sub>.

#### 9.3.1. Examinando um exemplo

Antes de escrever seu próprio arquivo de estilo, pode ser útil dar uma olhada em alguns dos arquivos padrão. Por exemplo, você pode abrir `book.ts` usando Arquivo→Carregar (não é necessário escolher um diretório, já que o diretório de estilos já se encontra no caminho de busca padrão).

Depois de abrir o arquivo `book.ts`, você verá muitas declarações de funções e ambientes (as declarações são visíveis já que os arquivos de estilo são escritos no “modo de preâmbulo (veja Opções→Modo)). Outras declarações estão nos arquivos `basic.ts`, `list.ts`, `theorem.ts` e `program.ts` nos quais baseia-se o arquivo `book.ts`. Estes arquivos contém, respectivamente, os ambientes básicos, para listas teoremas e programação.

#### 9.3.2. Diretórios importantes do T<sub>E</sub>X<sub>MACS</sub>

**Isto deveria estar em outro lugar.**

Para escrever seu próprio arquivo de estilo, é útil conhecer os seguintes diretórios importantes para do T<sub>E</sub>X<sub>MACS</sub> paths:

- `$TEXMACS_PATH` é o diretório principal do T<sub>E</sub>X<sub>MACS</sub>.
- `$TEXMACS_HOME_PATH` é o diretório principal do usuário para arquivos do T<sub>E</sub>X<sub>MACS</sub> (documentos, arquivos de estilo ou programas). Normalmente, este diretório é `~/TeXmacs`.

- `$TEXMACS_STYLE_ROOT` é a lista de diretórios nas quais são procurados arquivos de estilo. Normalmente, esta lista contém `$TEXMACS_PATH/styles` e `$TEXMACS_HOME_PATH/styles`.
- `$TEXMACS_PACKAGE_ROOT` é a lista de diretórios para pacotes de estilo. Normalmente, esta lista contém: `$TEXMACS_PATH/packages` e `$TEXMACS_HOME_PATH/packages`.
- `$TEXMACS_STYLE_PATH` é a lista de diretórios buscada para a inclusão de arquivos de estilo. Normalmente, esta lista contém `.` e todos os subdiretórios de `$TEXMACS_STYLE_ROOT` e `$TEXMACS_PACKAGE_ROOT`.
- `$TEXMACS_FILE_PATH` é a lista de diretórios para a busca de documentos. Normalmente, esta lista contém `$TEXMACS_PATH/texts` e `$TEXMACS_HOME_PATH/texts`.

## 9.4. ADAPTANDO OS ARQUIVOS E PACOTES PADRÃO DO T<sub>E</sub>X<sub>MACS</sub>

Sempre que os arquivos de estilo padrão do T<sub>E</sub>X<sub>MACS</sub> não são adequados para uma dada finalidade, é possível escrever seus próprios arquivos de estilo. Começar do zero, no entanto, pode ser muito complexo. Por isto, os arquivos de estilo do T<sub>E</sub>X<sub>MACS</sub> foram divididos em pacotes menores para facilitar o reaproveitamento de algumas partes deles. Eles também foram escritos de tal forma que você pode redefinir muitos macros *a posteriori*, o que permite que você adapte os arquivos de estilo mais facilmente.



# CAPÍTULO 10

## THE T<sub>E</sub>X<sub>MACS</sub> PLUG-IN SYSTEM

There are many ways in which T<sub>E</sub>X<sub>MACS</sub> can be customized or extended: users may define their own style files, customize the user interface, or write links with extern programs. The plug-in system provides a universal mechanism to combine one or several such extensions in a single package. Plug-ins are both easy to install by other users and easy to write and maintain.

### 10.1. INSTALLING AND USING A PLUG-IN

From the user's point of view, a plug-in *myplugin* will usually be distributed on some website as a binary tarball with the name

```
myplugin-version-architecture.tar.gz
```

If you installed T<sub>E</sub>X<sub>MACS</sub> yourself in the directory \$TEXMACS\_PATH, then you should unpack this tarball in the directory \$TEXMACS\_PATH/plugins, using

```
tar -zxvf myplugin-version-architecture.tar.gz
```

This will create a *myplugin* subdirectory in \$TEXMACS\_PATH/plugins. As soon as you restart T<sub>E</sub>X<sub>MACS</sub>, the plug-in should be automatically recognized. Please read the documentation which comes with your plug-in in order to learn using it.

**Comentário 10.1.** If you did not install T<sub>E</sub>X<sub>MACS</sub> yourself, or if you do not have write access to \$TEXMACS\_PATH, then you may also unpack the tarball in \$TEXMACS\_HOME\_PATH/plugins. Here we recall that \$TEXMACS\_HOME\_PATH defaults to \$HOME/.Texmacs. When starting T<sub>E</sub>X<sub>MACS</sub>, your plug-in should again be automatically recognized.

**Comentário 10.2.** If the plug-in is distributed as a source tarball like *myplugin-version-src.tar.gz*, then you should first compile the source code before relaunching T<sub>E</sub>X<sub>MACS</sub>. Depending on the plug-in (read the instructions), this is usually done using

```
cd myplugin; make
```

or

```
cd myplugin; ./configure; make
```

**Comentário 10.3.** In order to upgrade a plug-in, just remove the old version in \$TEXMACS\_PATH/plugins or \$TEXMACS\_HOME\_PATH/plugins using

```
rm -rf myplugin
```

and reinstall as explained above.

### 10.2. WRITING YOUR OWN PLUG-INS

In order to write a plug-in *myplugin*, you should start by creating a directory

```
$TEXMACS_HOME_PATH/plugins/myplugin
```

where to put all your files (recall that `$TEXMACS_HOME_PATH` defaults to `$HOME/.TeXmacs`). In addition, you may create the following subdirectories (when needed):

**bin** — For binary files.

**doc** — For documentation (not yet supported).

**langs** — For language related files, such as dictionaries (not yet supported).

**lib** — For libraries.

**packages** — For style packages.

**progs** — For SCHEME programs.

**src** — For source files.

**styles** — For style files.

As a general rule, files which are present in these subdirectories will be automatically recognized by T<sub>E</sub>X<sub>MACS</sub> at startup. For instance, if you provide a **bin** subdirectory, then

```
$TEXMACS_HOME_PATH/plugins/myplugin/bin
```

will be automatically added to the `PATH` environment variable at startup. Notice that the subdirectory structure of a plug-in is very similar to the subdirectory structure of `$TEXMACS_PATH`.

**Exemplo 10.4.** The easiest type of plug-in only consists of data files, such as a collection of style files and packages. In order to create such a plug-in, it suffices to create directories

```
$TEXMACS_HOME_PATH/plugins/myplugin
```

```
$TEXMACS_HOME_PATH/plugins/myplugin/styles
```

```
$TEXMACS_HOME_PATH/plugins/myplugin/packages
```

and to put your style files and packages in the last two directories. After restarting T<sub>E</sub>X<sub>MACS</sub>, your style files and packages will automatically appear in the Documento→Estilo and Documento→Usar pacote menus.

For more complex plug-ins, such as plug-ins with additional SCHEME or C++ code, one usually has to provide a SCHEME configuration file

```
$TEXMACS_HOME_PATH/plugins/myplugin/progs/init-myplugin.scm
```

This configuration file should contain an instruction of the following form

```
(plugin-configure myplugin
 configuration-options)
```

Here the *configuration-options* describe the principal actions which have to be undertaken at startup, including sanity checks for the plug-in. In the next sections, we will describe some simple examples of plug-ins and their configuration. Many other examples can be found in the directories

```
$TEXMACS_PATH/examples/plugins
```

```
$TEXMACS_PATH/plugins
```

Some of these are [described](#) in more detail in the chapter about writing new interfaces.

### 10.3. EXAMPLE OF A PLUG-IN WITH SCHEME CODE

#### The **world** plug-in.

Consider the world plug-in in the directory

```
$TEXMACS_PATH/examples/plugins
```

This plug-in shows how to extend  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  with some additional SCHEME code in the file

```
world/progs/init-world.scm
```

In order to test the world plug-in, you should recursively copy the directory

```
$TEXMACS_PATH/examples/plugins/world
```

to  $\$TEXMACS\_PATH/plugins$  or  $\$TEXMACS\_HOME\_PATH/plugins$ . When relaunching  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ , the plug-in should now be automatically recognized (a World menu should appear in the menu bar).

#### How it works.

The file `init-world.scm` essentially contains the following code:

```
(plugin-configure world
  (:require #t))

(when (supports-world?)
  (display* "Using world plug-in!\n"))
```

The configuration option `:require` specifies a condition which needs to be satisfied for the plug-in to be detected by  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  (later on, this will for instance allow us to check whether certain programs exist on the system). The configuration is aborted if the requirement is not fulfilled.

Assuming that the configuration succeeds, the `supports-world?` predicate will evaluate to `#t`. In our example, the body of the `when` statement corresponds to some further initialization code, which just sends a message to the standard output that we are using our plug-in. In general, this kind of initialization code should be very short and rather load a module which takes care of the real initialization. Indeed, keeping the `init-myplugin.scm` files simple will reduce the startup time of  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ .

### 10.4. EXAMPLE OF A PLUG-IN WITH C++ CODE

#### The **minimal** plug-in.

Consider the example of the minimal plug-in in the directory

```
$TEXMACS_PATH/examples/plugins
```

It consists of the following files:

```
minimal/Makefile
minimal/progs/init-minimal.scm
minimal/src/minimal.cpp
```

In order to try the plug-in, you first have to recursively copy the directory

```
$TEXMACS_PATH/examples/plugins/minimal
```

to `$TEXMACS_PATH/progs` or `$TEXMACS_HOME_PATH/progs`. Next, running the Makefile using

```
make
```

will compile the program `minimal.cpp` and create a binary

```
minimal/bin/minimal.bin
```

When relaunching T<sub>E</sub>X<sub>MACS</sub>, the plug-in should now be automatically recognized.

### How it works.

The `minimal` plug-in demonstrates a minimal interface between T<sub>E</sub>X<sub>MACS</sub> and an extern program; the program `minimal.cpp` is explained in more detail in the chapter about writing interfaces. The initialization file `init-minimal.scm` essentially contains the following code:

```
(plugin-configure minimal
  (:require (url-exists-in-path? "minimal.bin"))
  (:launch "minimal.bin")
  (:session "Minimal"))
```

The `:require` option checks whether `minimal.bin` indeed exists in the path (so this will fail if you forgot to run the Makefile). The `:launch` option specifies how to launch the extern program. The `:session` option indicates that it will be possible to create sessions for the `minimal` plug-in using `Inserir→Sessão→Minimal`.

## 10.5. SUMMARY OF THE CONFIGURATION OPTIONS FOR PLUG-INS

As explained before, the SCHEME configuration file `myplugin/progs/init-myplugin.scm` of a plug-in with name `plugin` should contain an instruction of the type

```
(plugin-configure myplugin
  configuration-options)
```

Here follows a list of the available *configuration-options*:

`(:winpath package-path inner-bin-path)` — Specify where to search for the plug-in under windows. The *package-path* is the usual place where the plug-in is installed. The *inner-bin-path* is the place where to look for the binary executable corresponding to the plug-in, relative to the *package-path*.

`(:winpath package-path inner-bin-path)` — Analogous to `:winpath`, but under MACOS.

- (`:require condition`) — This option specifies a sanity *condition* which needs to be satisfied by the plug-in. Usually, it is checked that certain binaries or libraries are present on your system. If the condition fails, then  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  will continue as whether your plug-in did not exist. In that case, further configuration is aborted. The `:require` option usually occurs first in the list of configuration options.
- (`:versions version-cmd`) — This option specifies a SCHEME expression *version-cmd* which evaluates to a list of available versions of the plug-in.
- (`:setup cmd`) — This command is only executed when the version of the plug-in changed from one execution of  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  to another one. This occurs mainly when installing new versions of  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  or helper applications.
- (`:launch shell-cmd`) — This option specifies that the plug-in is able to evaluate expressions over a pipe, using a helper application which is launched using the shell-command *shell-cmd*.
- (`:link lib-name export-struct options`) — This option is similar to `:launch`, except that the extern application is now linked dynamically. For more information, see the section about [dynamic linking](#).
- (`:session menu-name`) — This option indicates that the plug-in supports an evaluator for interactive shell sessions. An item *menu-item* will be inserted to the `Inserir→Sessão` menu in order to launch such sessions.
- (`:serializer ,fun-name`) — If the plug-in can be used as an evaluator, then this option specifies the SCHEME function *fun-name* which is used in order to transform  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  trees to strings.
- (`:commander ,fun-name`) — This command is similar to the `:serializer` option except that it is used to transform special commands to strings.
- (`:tab-completion #t`) — This command indicates that the plug-in supports tab-completion.
- (`:test-input-done #t`) — This command indicates that the plug-in provides a routine for testing whether the input is complete.

It should be noticed that the configuration of the plug-in *myplugin* automatically creates a few predicates:

*supports-myplugin?*. Test whether the plug-in is fully operational (all requirements are met).

*in-myplugin?*. Test whether *myplugin* is the current programming language.

*myplugin-scripts?*. Test whether *myplugin* is the current scripting language.



# CAPÍTULO 11

## RESUMO DAS PRINCIPAIS ETIQUETAS DO T<sub>E</sub>X<sub>MACS</sub>

### 11.1. A BASE COMUM PARA A MAIORIA DOS ESTILOS

A d.t.d. `common-base` contém a marcação que é comum a praticamente todos os estilos. Ela é dividida nas partes seguintes:

#### 11.1.1. Marcação Padrão

Uma variedades de marcações padrão são definidas em `std-markup`. Todas as etiquetas de conteúdo textual a seguir aceitam um argumento. A maioria pode ser encontrada no menu Inserir→Tag de conteúdo.

##### `strong`

Indica uma região **importante** do texto. Você pode introduzir esta etiqueta com Inserir→Tag de conteúdo→Forte.

##### `em`

Emfatiza um trecho do texto como em “a coisa *verdadeira*”. Esta etiqueta corresponde à entrada de menu Inserir→Tag de conteúdo→Destacar.

##### `dfn`

Para definições como “um *gnu* é um bicho cabeludo”. Esta etiqueta corresponde a Inserir→Tag de conteúdo→Definição.

##### `samp`

Uma seqüência de caracteres literais como ae a ligatura æ. Você pode inserir esta etiqueta com Inserir→Tag de conteúdo→Amostra.

##### `name`

O nome de alguma coisa coisa ou conceinto, como o sistema operacional LINUX. Esta etiqueta pode ser inserida com Inserir→Tag de conteúdo→Nome.

##### `person`

O nome de uma pessoa como JORIS. Esta etiqueta corresponde a Inserir→Tag de conteúdo→Pessoa.

##### `cite*`

Uma citação bibliográfica como um livro ou uma revista, por exemplo: *Moby Dick*, de Melville. Esta etiqueta, que encontra-se em Inserir→Tag de conteúdo→Citar, não deve ser confundida com `cite`. A última também é usada para citações, porém o argumento desta refere-se a uma entrada em um banco de dados de referências bibliográficas.

### abbr

Uma abreviação. Por exemplo, eu trabalho no C.N.R.S. Um abreviação é criada com Inserir→Tag de conteúdo→Abreviação ou com o atalho de teclado `\A`.

### acronym

Um acrônimo é uma abreviação formada com a primeira letra de cada palavra de uma frase, como HTML ou IBM. Em particular, as letras não separadas por pontos. Você pode inserir um acrônimo com Inserir→Tag de conteúdo→Acrônimo.

### verbatim

Texto literal como a saída de um programa de computador. Por exemplo: o programa disse: `hello`. Você pode digitar texto literal com Inserir→Tag de conteúdo→Verbatim. Esta etiqueta também pode ser usada como um ambiente para um trecho com vários parágrafos.

### kbd

Texto que dever ser digitado no teclado. Por exemplo: por favor tecele `return`. Esta etiqueta corresponde à entrada do menu Inserir→Tag de conteúdo→Teclado.

### code\*

Código fonte de um programa de computador, como em `cout << 1+1; imprime 2`". Esta etiqueta é inserida com Inserir→Tag de conteúdo→Código. Para trechos mais longos de código, você deve usar o ambiente `code`.

### var

Variáveis em um programa de computador, como `cp src-file dest-file`. Esta etiqueta corresponde à entrada de menu Inserir→Tag de conteúdo→Variável.

### math

Esta é a etiqueta que será usada no futuro para matemática dentro de texto normal. Por exemplo:  $\sin^2 x + \cos^2 x = 1$  é bem conhecida.

### op

Esta é a etiqueta que pode ser usada dentro de expressões matemáticas para indicar que um operador deve ser considerado por si só, sem qualquer argumento. Por exemplo: a operação  $+$  é uma função de  $\mathbb{R}^2$  em  $\mathbb{R}$ . Esta etiqueta pode tornar-se obsoleta.

### tt

É uma etiqueta de marcação física que existe para compatibilidade com HTML, mas não recomendamos seu uso. use.

A seguir listamos os ambientes padrão:

### verbatim

Descrito acima.

### code

Similar a `code*`, mas para trechos de código com várias linhas.

**quote**

Ambiente para citações curtas (um parágrafo).

**quotation**

Ambiente para citações longas (vários parágrafos).

**verse**

Ambiente para poesia.

**center**

Esta é uma etiqueta de marcação física para centralizar uma ou várias linhas de texto. Existe para compatibilidade com HTML, mas não recomendamos seu uso.

Alguns ambientes padrão para texto tabular são

**tabular\***

Tabelas centradas.

**block**

Tabelas alinhadas à esquerda com uma moldura padrão de 11n de largura.

**block\***

Tabelas centradas com uma moldura padrão de 11n de largura.

As etiquetas seguintes não aceitam argumentos:

**TeXmacs**

O logotipo do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ .

**TeX**

O logotipo  $\text{T}_{\text{E}}\text{X}$ .

**LaTeX**

O logotipo do  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

**hflush**

Usada por desenvolvedores para empurrar o texto para a direita na definição de ambientes.

**hrule**

Uma linha horizontal como a que aparece abaixo:

---

Todas as etiquetas abaixo aceitam um ou mais argumentos.

**overline**

Para grifado acima, que pode se estender por várias linhas.

**underline**

Para texto sublinhado, que pode se estender por várias linhas.

**fold**

Macro com dois argumentos. O primeiro é exibido e o segundo ignorado: este macro corresponde a apresentação “dobrada” de um trecho do documento associado com um breve título ou resumo. O segundo argumento pode ser mostrado com `Inserir→Switch→Unfold`.

**switch**

Macro com dois argumentos  $x$  e  $y$ , onde  $y$  é um conjunto de representações possíveis para a escolha e  $x$  a representação atual. As teclas de função `F9`, `F10`, `F11` e `F12` podem ser usadas para alternar entre as diferentes representações.

**phantom**

Função com um argumento  $x$ . Esta etiqueta ocupa tanto espaço quanto o argumento  $x$  ocuparia quando tipografado, porém  $x$  não é exibido. Por exemplo, o texto “fantasma” como um argumento para `phantom` fornece “ ”.

**set-header**

Função com um argumento para mudar de forma permanente o cabeçalho. Perceba que algumas etiquetas no arquivo de estilo, como etiquetas de sessão, podem sobrepor-se a esta definição manual.

**set-footer**

Função com um argumento para alterar de forma permanente o rodapé.

**11.1.2. Símbolos padrão**

A d.t.d. `std-symbol` define os símbolos especiais  $\text{¢}$ ,  $\text{¤}$ ,  $\text{¥}$ ,  $\text{©}$ ,  $\text{®}$ ,  $\text{°}$ ,  $\text{²}$ ,  $\text{³}$ ,  $\text{¹}$ ,  $\mu$ ,  $\text{¶}$ ,  $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{3}{4}$ ,  $\text{€}$  e  $\text{™}$ . Assim que o funcionamento do sistema de fontes seja melhorado, esta d.t.d. tornar-se-á obsoleta.

**11.1.3. Marcação matemática padrão**

A marcação matemática padrão é definida em `std-math`.

**binom**

Para coeficientes binomiais  $\binom{n}{m}$ .

**choose**

Outro nome para `binom` (obsoleto).

**shrink-inline**

Um macro que muda o tamanho do texto para “scriptsize” quando você não está no estilo de exibição. Este macro é usado principalmente pelos desenvolvedores. Por exemplo, é usado em `binom`.

Abaixo estão os ambientes matemáticos padrão:

**matrix**

Para matrizes  $M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ .

`det`

Para determinantes  $\Delta = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$ .

`choice`

Para listas de escolhas  $|x| = \begin{cases} -x, & \text{if } x \leq 0 \\ x, & \text{if } x \geq 0 \end{cases}$

#### 11.1.4. Listas padrão

As listas padrão do  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  são definidas em `std-list`. As listas sem numeração são:

`itemize`

A marca antes de cada item depende do nível do aninhamento.

`itemize-minus`

Usa `–` para a marca.

`itemize-dot`

Usa `•` para a marca.

`itemize-arrow`

Usa `→` para a marca.

Listas numeradas correspondem aos seguintes ambientes:

`enumerate`

O tipo do número depende do nível de aninhamento.

`enumerate-numeric`

Numera os itens com 1, 2, 3, etc.

`enumerate-roman`

Numera os itens com i, ii, iii, etc.

`enumerate-Roman`

Numera os itens com I, II, III, etc.

`enumerate-alpha`

Numera os itens com a), b), c), etc.

`enumerate-Alpha`

Numera os itens com A, B, C, etc.

Os ambientes abaixo são usados para listas descritivas.

`description`

O ambiente padrão para listas descritivas (normalmente `description-compact`).

`description-compact`

Alinha os itens da lista à esquerda e coloca as descrições imediatamente depois deles.

### description-dash

Parecido com [description-compact](#), mas usa um — para separar cada item de sua descrição.

### description-align

Alinha as descrições à esquerda, enquanto que os itens são alinhados à direita.

### description-long

Coloca os itens e suas descrições em linhas diferentes.

Novos itens em uma lista são indicados através da etiqueta [item](#) ou da etiqueta unária [item\\*](#), no caso de descrições. Desenvolvedores podem encontrar ainda outras macros em [std-list](#) para definir outras estruturas de listas, porém estes macros não são considerados estáveis.

## 11.1.5. Geração de conteúdo automática

A d.t.d. [std-automatic](#) contém definições para a geração de conteúdo automático como índices e bibliografia assim como para a apresentação deste material. As etiquetas abaixo são usadas para bibliografias:

### cite

Uma função com um número arbitrário de argumentos. Cada argumento é uma citação correspondente a um item em um arquivo BiB-T<sub>E</sub>X. As citações são mostradas da mesma forma em que são referenciadas na bibliografia e elas também fornecem hiperligações para as referências correspondentes. As citações são mostradas como pontos de interrogação se você não gerou a bibliografia.

### nocite\*

Semelhante a [cite](#), mas as citações não são exibidas no texto principal.

### bibitem\*

Uma função que especifica como exibir um texto na bibliografia.

As etiquetas abaixo são usadas para compilar um índice:

### toc-main-1

Uma função com um argumento para criar uma entrada inicial no índice. Esta função pode por exemplo ser usada quando um livro é composto de várias partes.

### toc-main-2

Uma função com um argumento para criar uma entrada principal no índice. Esta função normalmente é usada para capítulos.

### toc-normal-1

Uma função com um argumento para criar uma entrada normal no índice. Esta função normalmente é usada para sessões.

### toc-normal-2

Semelhante a [toc-normal-2](#), para entradas menos importantes como subseções.

### toc-normal-3

Semelhante a `toc-normal-3`, para entradas ainda menos importantes.

### toc-small-1

Usada para entradas pouco importantes, tais como parágrafos (podem ser ignoradas.)

### toc-small-2

Usada para entradas ainda menos importantes do que `toc-small-1`, como subparágrafos.

### toc-dots

A separação entre uma entrada do índice e o número da página correspondente. Normalmente, são usados pontos horizontais.

As seguintes etiquetas são usadas para índices remissivos (no final do documento):

### index

Uma função com um argumento  $x$ , que insere  $x$  no índice como uma entrada principal.

### subindex

Uma função com dois argumentos,  $x$  e  $y$ , que insere  $y$  no índice como uma entrada subordinada a  $x$ .

### subsubindex

Uma função com três argumentos,  $x$ ,  $y$  e  $z$ , que insere  $z$  no índice como uma entrada subordinada a  $y$ , que é subordinada a  $x$ .

### index-complex

Uma função com quatro argumentos, *key*, *how*, *range*, *entry*, que é documentada na sessão sobre *geração de índices*.

### index-line

Esta função aceita uma *chave* como argumento, que diz como ordenar a entrada, e a *entry* em si. Não é gerado um número de página.

### index-1

Macro com uma entrada no índice e um número de página, que é usado para exibir uma entrada principal do índice.

### index-1\*

Semelhante a `index-1`, mas sem o número da página.

### index-n

(com  $n$  entre 1 e 5): macro com uma entrada no índice e um número de página, que é usada para apresentação de uma entrada do índice de nível  $n$ .

### index-n\*

Semelhante a `index-n`, mas sem o número da página.

### index-dots

O macro que produz os pontos entre as entradas do índice e os números de página correspondentes.

As etiquetas abaixo são usadas para glossários:

### glossary

Uma função que insere seu argumento único no glossário.

### glossary-dup

Para criar um número de página adicional para uma entrada que já tinha sido inserida no glossário.

### glossary-explain

Uma função para inserir uma entrada do glossário com sua explicação.

### glossary-line

Inserir uma entrada no glossário sem um número de página.

### glossary-1

Macro para apresentação de uma entrada do glossário com o seu número de página correspondente.

### glossary-2

Macro para apresentação de uma entrada do glossário, sua explicação, e seu número de página.

### glossary-dots

O macro que produz os pontos entre a entrada no glossário e seu número de página correspondente.

## 11.1.6. Marcação especial para programas e sessões interativas

A d.t.d. `program` fornece principalmente os seguintes ambientes para álgebra computacional:

### session

Macro com três argumentos: a linguagem de álgebra computacional, o nome da sessão e o próprio corpo da sessão.

### input

Macro com dois argumentos: um prompt e a própria entrada.

### output

Macro com o corpo da saída como seu argumento.

Na verdade, estes ambientes estão baseados em ambientes da forma `lan-session`, `lan-input` e `lan-output` para cada linguagem `lan`.

A d.t.d. `program` também fornece alguma marcação para diagramação de programas de computador. Estas etiquetas devem ser consideradas, no entanto, muito voláteis, já que pretendemos substituí-las por um conjunto de etiquetas bem mais detalhado:

#### `algorithm`

Macro com dois argumentos, o nome do algoritmo e o algoritmo em si, possivelmente acoplado à sua especificação.

#### `body`

O corpo do algoritmo.

#### `indent`

Para recuar parte do algoritmo.

## 11.2. AMBIENTES PADRÃO PARA TEXTO

A d.t.d. `env` contém os ambiente padrão disponíveis na maioria dos estilos. Ela é subdividida nas seguintes partes:

### 11.2.1. Definição de novos ambientes

A d.t.d. `env-manage` contém marcação de alto nível que pode ser usada pelo usuário para definir novos ambientes para teoremas, comentários, exercícios e figuras:

#### `new-theorem`

Define um ambiente similar a um teorema. Você deve especificar o nome para o ambiente (como “experiência”) e o texto correspondente (como “Experiência”).

#### `new-remark`

Semelhante a `new-theorem`, mas para comentários.

#### `new-exercise`

Semelhante a `new-theorem`, mas para exercícios.

#### `new-figure`

Semelhante a `new-theorem`, mas para figuras (em pares para figuras grandes e pequenas).

Esta d.t.d. também contém marcação de baixo nível para a definição destes ambientes. Na verdade, a definição de novos ambientes para teoremas é feita em duas etapas. Na primeira, a etiqueta `new-theorem` é usada para especificar qual ambiente deve ser definido. Na segunda etapa, (imediatamente antes do documento do usuário ser definido), os ambientes são efetivamente definidos. Este mecanismo torna possível modificar os ambientes com pacotes que são processados entre estas duas etapas. Por exemplo, a numeração dos teoremas é modificada desta forma.

**Aviso 11.1.** No momento, você deve usar `new-theorem` e etiquetas similares apenas dentro de arquivos de estilo ou pacotes pessoais. Se você usar `new-theorem` diretamente dentro de um documento, a numeração poderá ser incorreta, devido ao esquema com duas etapas descrito acima. Esta limitação irá desaparecer tão logo seja possível especificar preâmbulos limpos para documentos  $\text{\TeX}_{\text{MACS}}$ .

### 11.2.2. Ambiente matemáticos

A d.t.d. `env-math` especifica os ambiente matemático que podem ser usados no modo texto. Em outras palavra, os ambientes devem ser usados no modo texto, porém seus corpos contém fórmulas matemáticas ou tabelas de fórmulas matemáticas.

#### `equation`

Uma equação numerada.

#### `equation*`

Uma equação sem número.

#### `eqnarray`

Uma matriz de equações numeradas (não deve ser usada ainda).

#### `eqnarray*`

Uma matriz de equações não numeradas.

Dentro do ambiente `eqnarray*`, você pode usar a etiqueta `eq-number` para numerar a equação

**Aviso 11.2.** A numeração de equações dentro de tabelas ainda não é exatamente como deveria. Em particular, a etiqueta `eqnarray` é equivalente a `eqnarray*` no momento. Mais tarde, quando a etiqueta `eqnarray` for implementada corretamente, você também terá uma etiqueta `no-number` para suprimir a numeração de uma equação, e um arquivo de estilo para numerar as equações do lado esquerdo.

**Aviso 11.3.** Não há ainda uma opção para numerar as equações do lado esquerdo da página. Ainda assim, você pode manualmente usar a etiqueta `leq-number` para obter o mesmo efeito. Também existe a etiqueta `next-number` que mostra o próximo número e incrementa o contador de equações.

**Aviso 11.4.** Nós não encorajamos o uso dos ambientes do AMS-T<sub>E</sub>X `align`, `gather` e `split`. Ainda assim, eles estão disponíveis sob os nomes de `align`, `gather`, `eqsplit`, juntamente com suas variantes `align*`, `gather*` e `eqsplit*`. Nós planejamos fornecer no futuro ambientes mais poderosos.

### 11.2.3. Ambientes para teoremas e afins

A d.t.d. `env-theorem` fornece marcação para a diagramação de teoremas e similares. As mais importantes etiquetas são:

#### `render-theorem`

Um macro para formatar ambientes similares a teoremas. O primeiro argumento especifica o nome do teorema, algo como “Teorema 1.2” e o segundo argumento contém o corpo do teorema. Este ambiente é usado para teoremas definidos por `new-theorem`.

#### `render-remark`

Semelhante a `render-theorem`, mas usado para ambientes similares a comentários.

#### `render-exercise`

Semelhante a `render-theorem`, mas para ambientes semelhantes a exercícios.

### render-proof

Semelhante a [render-theorem](#), mas para provas. Este ambiente é usado principalmente para alterar o nome da prova, como em “Fim da prova do teorema 1.2”

### dueto

Um ambiente para especificar os criadores de um teorema.

### corollary\*

Para corolários não numerados. Este ambiente é baseado em [render-theorem](#).

### proof

Para provas de teoremas. Este ambiente é baseado em [render-proof](#).

As etiquetas seguintes podem ser usadas para modificação dos ambientes.

### theorem-name

Um macro que controla a aparência dos nomes dos ambiente para teoremas e comentários. A maioria dos estilos usa negrito ou maiúsculas pequenas.

### exercise-name

Semelhante a [theorem-name](#), mas para exercícios.

### theorem-sep

O separador entre o nome do teorema ou similar e seu corpo. Em geral, um ponto seguido de um espaço.

### exercise-sep

Semelhante a [theorem-sep](#), mas para exercícios.

## 11.2.4. Ambientes para objetos flutuantes

A d.t.d. [env-float](#) fornece marcação para objetos flutuantes. A etiqueta seguinte é a única de alto nível.

### footnote

Cria uma nota de rodapé.

As etiquetas de baixo nível a seguir podem ser usadas para a definição de ambientes de alto nível para figuras e tabelas, como [big-figure](#), [small-figure](#), [big-table](#) e [small-table](#):

### render-small-figure

Um macro para exibição de uma figura pequena. Os argumento são um nome curto (como “figura” ou “tabela”) para a lista de figuras, seu nome verdadeiro (como “Figura 2.3” ou “Tabela 5”), a própria figura e um texto para a legenda.

### render-big-figure

Uma variação de [render-small-figure](#) para exibição de figuras grandes.

As etiquetas abaixo podem ser usadas para modificar a aparência do texto em torno das figuras, tabelas e notas de rodapé.

**figure-name**

Um macro que controla a aparência do texto“Figura”. O padrão é usar negrito.

**figure-sep**

O separador entre a figura e seu número e a legenda. O padrão é um ponto seguido por um espaço.

**footnote-sep**

O separador entre o número de uma nota de rodapé e o texto. O padrão é um ponto seguido por um espaço.

### 11.2.5. Ambientes padrão

A d.t.d. `env-default` contém os ambientes textuais padrão. Eles estão divididos nos grupos seguintes:

**Variantes de teoremas.** Os corpos de ambientes de teoremas e similares são geralmente enfatizados. Em uma instalação padrão, os seguintes ambientes estão disponíveis no menu Inserir→Ambiente: `theorem`, `proposition`, `lemma`, `corollary`, `axiom`, `definition`, `notation`, `conjecture`.

**Variantes de comentários.** Os seguintes ambientes estão disponíveis através do menu: Inserir→Ambiente: `remark`, `example`, `note`, `warning`, `convention`.

**Variantes de exercícios.** O menu Inserir→Ambiente fornece dois ambientes assim: `exercise` and `problem`.

**Variantes de figuras.** Estes ambientes sempre vêm aos pares, grandes e pequenos. Uma instalação padrão fornece `big-figure`, `small-figure`, `big-table` e `small-table`. Você pode encontrá-los em Inserir→Imagem e Inserir→Tabela.

**Outros ambientes úteis.** Fornecemos também `keywords` e `AMS-class` (para a classificação por assuntos da A.M.S.). Estes ambientes devem ser usados apenas dentro de um `abstract`.

## 11.3. HEADERS AND FOOTERS

### 11.4. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ STYLE SECTIONS

The `section-latex` d.t.d. provides the standard tags for sections, which are the same as in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Most sectional tags take only one argument: the name of the section. In the future, we plan to provide alternative tags with two arguments, which will allow you to see the body of a section as part of the structure. The following tags usually yield numbered sections, which are referenced in the table of contents:

### chapter

Macro for producing a potentially numbered chapter title.

### section

Macro for producing a potentially numbered section title.

### subsection

Macro for producing a potentially numbered subsection title.

### subsubsection

Macro for producing a potentially numbered subsubsection title.

### paragraph

Macro for producing a potentially numbered paragraph title.

### subparagraph

Macro for producing a potentially numbered subparagraph title.

The tags `chapter*`, `section*`, `subsection*`, `subsubsection*`, `paragraph*` and `subparagraph*` can be used for producing the unnumbered variants of the above tags, which are not referenced in the table of contents. The `section-latex` d.t.d. also provides the following tags:

### chapter\*\*

Macro with two arguments: a special type of chapter (like “Epilogue”) and the name of the chapter.

### appendix

A variant of `chapter` or `section` for producing appendices.

### section-sep

A macro for customizing the separator between the number of a section and its title. By default, we use two spaces.

## 11.5. MARKUP FOR AUTOMATIC NUMBERING



# ÍNDICE

Ajuda	
Scheme	34
Algoritmo	31
Arquivo	
Carregar	9–10, 38
Exportar	
Postscript	10
Imprimir	
Imprimir tudo	10
Imprimir tudo em um arquivo	10
Novo	10
Salvar	10
Salvar como	10
<code>common-base</code>	47
Documento	9
Atualizar	
Bibliografia	28
Tudo	28
Estilo	10–11, 37, 42
Fonte	11
Dpi	10
Tamanho	15
Linguagem	10–11
Mestre	
Vincular	29
Página	14
Layout	14
Layout da tela	14
Margens como no papel	10
Partindo	31
Tamanho	10
Tipo	14, 28
Papel	10
Usar pacote	37, 42
Programa	36
Editar	
Colar	33
Copiar	33
Copiar para	33
Cortar	33
Desfazer	34
Exportar	33
Importar	33
Ortografia	34
Preferências	9, 15
Procurar	33
Refazer	34
Substituir	33
<code>env</code>	55
<code>env-default</code>	58
<code>env-float</code>	57
<code>env-manage</code>	55
<code>env-math</code>	56
<code>env-theorem</code>	56
Ferramentas	
Atualizar	
Inclusões	29
Flexibilidade	31
Formato	
Cor	
Vermelho	33
Espaço	14
Inserção de página	
Figura flutuante	31
Nota de rodapé	31
Objeto flutuante	31
Tabela flutuante	31
<code>framed-session</code>	36
Inserir	
Ambiente	11, 13, 58
Automático	
Índice	28
Bibliografia	28
Tabela de conteúdos	28
content tags	12
Descrição	13
Enumerar	11, 13
Roman	13
Formato da fonte	
Itálico	9
Fração	33
Imagem	27, 58
Figura pequena	31
Link	
Ação	27
Citação	28
Citação invisível	28
Etiqueta	27
Hiperlink	27
Incluir	27, 29
Item do índice	28–29
Referência	27
Listar	11–12
Matemáticos	
Equação	19
Equações	23
Posicionar flutuante	31
Seção	11
Sessão	35, 45
Minimal	44
Outro	35
Switch	
Unfold	50
Tabela	23, 58

Tabela pequena . . . . .	31	Insert fields . . . . .	36
Tag de conteúdo . . . . .	47	Fold input field . . . . .	36
Abreviação . . . . .	48	Insert text field . . . . .	36
Acrônimo . . . . .	48	Interromper execução . . . . .	35
Amostra . . . . .	47	Remove fields . . . . .	36
Código . . . . .	48	Remove all output fields . . . . .	36
Citar . . . . .	47	Split session . . . . .	36
Definição . . . . .	47	<code>std-automatic</code> . . . . .	52
Destacar . . . . .	47	<code>std-list</code> . . . . .	51–52
Forte . . . . .	47	<code>std-markup</code> . . . . .	47
Nome . . . . .	47	<code>std-math</code> . . . . .	50
Pessoa . . . . .	47	<code>std-symbol</code> . . . . .	50
Teclado . . . . .	48	Tabela	
Variável . . . . .	48	Alinhamento horizontal da célula . . . . .	24
Verbatim . . . . .	48	Alinhamento horizontal da tabela . . . . .	24
Ir para . . . . .	10	Alinhamento vertical da célula . . . . .	24
Limites . . . . .	31	Alinhamento vertical da tabela . . . . .	24
Opções		Altura da célula	
Modo . . . . .	38	Definir altura . . . . .	24
Segurança . . . . .	27	Borda da célula . . . . .	24
Parágrafo		Cor de fundo da célula . . . . .	25
Número de colunas . . . . .	31	Largura da célula	
Preferências		Definir largura . . . . .	24
Impressora . . . . .	10	Modo de operação da célula . . . . .	23
<code>program</code> . . . . .	54–55	Propriedades especiais de célula	
<code>section-latex</code> . . . . .	58–59	Distribuir espaço vazio . . . . .	24
Sessão		Propriedades especiais de tabela	
Encerrar sessão . . . . .	35	Borda . . . . .	25
Input mode		Extrair formato . . . . .	25
Mathematical input . . . . .	36	Texto . . . . .	11
Multiline input . . . . .	36	World . . . . .	43