

O compilador de documentos L^AT_EX

Praciano-Pereira, T.*

Sobral Matemática
Universidade Estadual Vale do Acaraú

7 de maio de 2012

tarcisio@member.ams.org
préprints da Sobral Matemática
no. 2012.02

Editor Tarcisio Praciano-Pereira
tarcisio@member.ams.org

Resumo

Este artigo descreve num formato de tutorial o que é o L^AT_EX, um compilador de documentos. O objetivo é abrir a perspectiva do seu uso para professores do Ensino Médio tanto para oferecer-lhes um instrumento que pode agilizar e produzir melhores textos, como também para que eles divulguem L^AT_EX com seus estudantes. Uma das virtudes deste programa é ajudar na organização do trabalho permitindo construir diversos textos a partir de uma base de dados de textos.

A história do L^AT_EX irá ficar muito curta, apenas mencionada, porque sobre este assunto existe muita coisa já escrita, vou me limitar a apontar referências.

palavras chave: L^AT_EX, compilador de textos, reutilização de textos.

This paper is about L^AT_EX, a program to compile texts. It is written as a tutorial and the goal is to reach the teacher of the High School with twofold objective one of them being to offer them the information about this tool that can make their life easier regarding the production of documents and other being to use them as vector to reach their students. One of the virtues of this program is to make it possible to produce several different texts from a data base of texts the author have build.

The history of L^AT_EX will no be attempted here as there is much written about the program, I will limit myself to point references about it.

keywords: L^AT_EX, texts compiler, texts reusing.

*U Vale do Acaraú - Sobral - Brazil - tarcisio@member.ams.org

1 Plano do trabalho

Esta seção introdutória tem dois objetivos, um dizer o que é \LaTeX para alguém que porventura nunca tenha lido nada a respeito, e também descrever o plano do trabalho.

\LaTeX é um pacote que funciona em cima de do \TeX , sendo conhecido como um pacote de macros que torna fácil a aplicação do primitivo \TeX que foi escrito nos anos 70 por Donald Knuth com o objetivo ter uma linguagem de programação para escrever um livro, *The Art of Computer Programming*, que teria 7 volumes mas que até agora não passou de quatro. Pode parecer que estou tentando difamar Knuth, o que seria uma grande bobagem, porque apenas a construção do \TeX bastaria para garantir a sua fama e ele fez bem mais do isto. Se quiser ter uma pequena visão dos seus feitos vá até [5], porém, de qualquer forma, ele não conseguiu até hoje terminar o livro que era seu objetivo quando escreveu \TeX e na minha opinião já perdeu sentido.

Não seria possível ir a fundo sobre o que é \LaTeX num texto do tamanho de um artigo, e o que pretendo é falar de sua existência para os que nada sabem a respeito e terei preenchido meu objetivo se conseguir atizar a curiosidade da leitora para aprender mais. Se quiser saber mais, basta procurar na rede e vai encontrar o mundo do \LaTeX , deverá ter o cuidado de selecionar o que ler para conseguir aprender e não se assustar, vou explicar o porque desta observação.

\LaTeX é um programa para compilar textos, no início, quem começa a aprender, tem uma sensação de que está se metendo numa grande complicação, foi esta a minha sensação quando tomei conhecimento de sua existência. Para começar teria que usar comandos para obter efeitos interessantes no texto. Isto hoje não é mais verdade porque há pacotes gráficos que oferecem todo um sistema de ajuda, com botões, e você pode usar os comandos do \LaTeX e continuar ignorando que eles existem.

Para não perder a oportunidade, o símbolo \LaTeX é produzido por um comando do \LaTeX . A figura (1), página 1, mostra como é o comando e como eu o redefini para ficar mais fácil para o meu uso. Fica logo aqui este primeiro exemplo de como podemos modificar o \LaTeX com os nossos próprios comandos. Na seção 4 eu vou mostrar-lhe como podemos definir novos comandos do \LaTeX , ou mesmo redefinir os comandos existentes.

Deixe-me aproveitar o ensejo para observar que inseri uma figura no texto, e que o fiz colocando texto em volta da figura. Com isto eu posso economizar papel e escrever textos mais bonitos.

Passado o primeiro susto você descobre que \LaTeX é um compilador de textos com um poderio fora do comum. E se aprofundando um pouquinho poderá descobrir a sua grande capacidade que continua

Este é o comando:

```
\LaTeX  
Esta é a minha  
definição:  
\latex
```

Figura 1: Redefini assim o comando \LaTeX

sendo aumentada pelos acréscimos que uma quantidade incontável de desenvolvedores fazem adicionando novos efeitos ou possibilidades frequentemente, e isto será um dos principais feitos de Knuth uma vez que ele é pioneiro ao convocar os outros para que se venham juntar ao seu trabalho com o desafio de US\$1,00 que fez para quem encontrasse um erro no livro que escreveu sobre $\text{T}_{\text{E}}\text{X}$, [6].

Somente para dar duas ideias, (1) há editoras de grande porte como *Academic Press*, um exemplo, que automatizam toda a sua produção de textos com este programa, e são centenas de editoras que o fazem, (2) é capaz de se comunicar com linguagens de programação portanto podemos com ele automatizar vários processos de edição de textos. Vou basear este artigo em um pequeno exemplo, um modelo para produzir listas de exercícios, que pode ser facilmente adaptado para gerenciar um sistema de correspondências, como de um sindicato, duma coordenação de curso, ou de uma sociedade científica ou literária.

Possivelmente $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ é a primeira *linguagem de marcação* (markup language) seguida do HTML e hoje é possível traduzir um documento de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ para HTML e vice versa. Quer dizer, o seu texto em $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ está pronto para publicar na web.

O que é uma *linguagem de marcação*? é o nome que se dá as linguagens de programação destinadas a produzirem textos. $\text{T}_{\text{E}}\text{X}$ produz textos para serem impressos em papel, HTML e seus derivados produzem texto para serem apresentados na tela do computador ou melhor, dentro de uma página da Internet. Ainda tem que faça uso do diretamente do $\text{T}_{\text{E}}\text{X}$, mas a grande maioria dos usuários prefere $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ que é o assunto deste artigo.

Deixe-me agora apresentar-lhe o plano deste trabalho para que você aquilate as limitações do meu trabalho e entenda como poderá ir mais longe.

- Esta primeira seção serviu para apresentar o sistema, como ele surgiu e quem se encontra à raiz de sua história.
- A segunda seção vai lhe mostrar um documento simples, com o objetivo de lhe provar que você poderá começar a usar $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ assim que terminar de ler este artigo.
- Na terceira seção vou lhe mostrar como eu uso $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ para escrever documentos mais complexos, um livro ou um artigo como este. Mas estou lhe mostrando a minha opção pessoal e muita gente usa $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ dentro de sistemas gráficos que fazem, quase tudo que eu faço manualmente, com uso de botões. Quase tudo, fique bem claro! Se você quiser *dar nó em pingo d'água* com $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, será melhor abrir a *caixa preta* e aprender a usar os comandos. Leia o *livro do $\text{T}_{\text{E}}\text{X}$* , [6], você irá encontrá-lo fazendo uma busca na Internet. Tem uma versão livre, em **pdf** que pode ser baixada. e se você ainda não viu, baixe a versão livre, em **pdf** para ver o que é possível fazerem-se com textos usando $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Também é comum que os autores usem primeiro uma interface gráfica e depois entrem no *código fonte* para melhorar alguns detalhes.

A terceira seção vai se basear num exemplo, a construção de uma lista de exercícios que pode lhe servir de modelo para diversos outros tipos de documentos simples mas *high tech...* como cartas automáticas de um grupo ou uma sociedade técnica ou literária.

- A quarta seção será muito curta apenas dando alguns exemplos avançados para personalização de documentos, vou entrar em alguns detalhes técnicos, como uso de banco de dados bibliográficos, uso de variáveis para fazer referências internas dentro de um texto e como criar um *índice remissivo alfabético*, como este que você pode encontrar ao final deste artigo. Vou deixar à sua disposição alguns *códigos fonte* que poderão lhe ser útil caso você se decida por usar \LaTeX , como eu espero.
- As referências bibliográficas tem como objetivo apresentar-lhe uma lista de referências nas quais você poderá iniciar o seu aprendizado ou aprofundá-lo. Neste sentido ela foge às regras bibliográficas por que há itens que não serão citados no texto.

2 Um documento inicial

Optei por apresentar duas figuras, a primeira figura (2), página 14, mostra o código latex mais simples possível.

Na segunda figura (3), página 15, você pode ver o resultado, uma foto do pdf depois da compilação.

Vou fazer comentários explicativos do *código fonte*, compare com o resultado final, na figura (3).

Os comandos em \LaTeX , se caracterizam pelo uso da contrabarra.

1. As três primeiras linhas definem o documento.
 - (a) O comando `\documentclass` recebe um parâmetro que pode ter um dos seguintes valores: `report`, `book`, `article`, `letter` que se traduzem em português, respectivamente, por *relatório técnico*, *livro*, *artigo*, *carta*. Há outros parâmetros opcionais que servem para selecionar o tipo de letra, o tipo de papel. Os comandos do \LaTeX são caracterizados pela contra-barra, `\`, usam as *chaves* `{`, `}` para receber parâmetros obrigatórios e *colchetes*, `[`, `]`, para os parâmetros opcionais. Se você não quiser usar parâmetros opcionais, pode deixar os colchetes vazios, ou simplesmente omití-los. Sugiro que deixe vazios para que se lembre que é possível usar parâmetros opcionais. \LaTeX não imprime as chaves, mas reclama de erro se não as encontrar em duplas, abrindo e fechando. Para imprimir *chaves*, sinal de *percentual*, o sinal do *jogo da velha*, use estes sinais antecedi-dos da contra-barra. Por exemplo, estou imprimindo estes sinais: `{`, `#`, `%`. Isto, porquê todos estes símbolos tem significado especial para \LaTeX . Toda vida que você quiser imprimir um símbolo espacial, anteceda-o da contra-barra.

- (b) O comando `\usepackage[utf8]{inputenc}` serve para inserir uma biblioteca, neste caso para definir os caracteres da lingua brasileira. Sem este comando não é possível com a nossa grafia.
 - (c) O comando `\usepackage[brazil]{babel}` complementa a definição com hifenação.
2. A quarta linha marca o início do documento e é finalizado com a última linha `\end{document}`. É prudente ao iniciar um documento colocar logo este par: `\begin{document} \end{document}`. Se você esquecer o `LATEX` não irá descobri-lo mas irá parar dizendo-lhe que excedeu a capacidade de memória, possivelmente. É um dos erros mais difíceis de se descobrir.

Entre estes dois comandos, fica o texto que não precisa ter nenhum outro comando se for texto puro. Quando você saltar uma linha, `LATEX` irá compreender que deve iniciar um novo parágrafo.

Neste texto tem o comando para produzir símbolo do `LATEX`. Se você precisar de espaço depois do comando, use a contra barra que é o comando para produzir um espaço. Se não houver este comando para produzir espaço, o símbolo do `LATEX` irá se juntar com a próxima palavra. Isto não é erro, é que depois do símbolo poderia vir uma vírgula, ou um outro sinal de texto. A autora deve decidir se deseja espaço ou não. Isto vale para qualquer comando do `LATEX`. Aproveito o ensejo, para falar do comando para inserir uma linha em branco, é a dupla contra-barra, `\\`.

Também tem o comando para produzir uma *lista numerada* dentro do qual aparecem os comandos `item` que marcam cada um dos itens da lista.

O comando `emph` é para enfatizar o seu parâmetro “Academic Press”.

Com isto você já pode escrever uma carta simples sem grandes efeitos. Para escrever textos simples você agora já conhece tudo. Resta apenas informá-lo como obter o resultado final.

2.1 Como obter o resultado final

Primeiro que tudo você tem que instalar o `LATEX`. Se for um usuário de Linux, é trivial, sua distribuição tem o programa, apenas você precisa instalá-lo. Uma das versões mais comuns se chama `tex-live` (ou talvez `texlive`). Experimente. Se você usa outro sistema eu não tenho condições de ajudá-la por desconhecer totalmente qualquer coisa que não seja Linux.

Vou supor que você tem `LATEX` instalado, que já escreveu o seu primeiro texto, e que você gravou o arquivo com o nome: “*primeiro.tex*”. No diretório em que você gravou o arquivo, digite, num terminal:

```
pdflatex primeiro.tex
```

Se houver erros, o `LATEX` vai informá-la disto. Se não houver erros o resultado será um arquivo com o nome “`primeiro.pdf`”. Use o leitor de pdf para ver o texto na tela. No meu caso eu faria:

```
xpdf primeiro.pdf
```

O leitor de **pdf** tem a opção de impressão, se você quiser imprimir a sua primeira carta em L^AT_EX.

3 Lista de exercícios

Se você for trabalhar com um documento mais complexo, que tenha capítulos ou seções é prudente criar um arquivo-mestre. Eu tenho um arquivo-mestre padrão que chamo de *rosto.tex*, pronto para produzir textos em português. A figura (4) página 16, lhe mostra o conteúdo do arquivo *rosto.tex*. Se você tentar compilar este arquivo, L^AT_EX vai aceitar sem problemas e irá produzir um documento vazio. Observe que entre `\begin{document}` e `\end{document}` não há nada. Mas se você executar as duas operações seguintes

- Do documento que aparece na figura (2), página 14, que suponho que você o tenha gravado com o nome “*primeiro.tex*”, retire `\begin{document}` e tudo que vier antes. Apague `\end{document}`. Grave com o nome “*segundo.tex*” para não perder “*primeiro.tex*”, no caso em que o deseje preservar.
- Em *rosto.tex*, entre `\begin{document}` e `\end{document}` inclua:

```
\input{segundo}
```

Observe: não é *segundo.tex*, é apenas *segundo*. Se compilar e depois acionar o visualizador de **pdf** verá o mesmo resultado que se encontra na figura (3), página 15.

Para não perder a ocasião deixe-me chamar sua atenção que estou fazendo referências, dentro deste documento, às figuras que se encontram em outras páginas do documento. Estou me referindo às figuras e às páginas onde elas se encontram.

Aqui você está vendo o método para produzir um documento que tenha vários capítulos, ou várias seções. Cada parte do documento incluída com um comando `\input{ }`. Cada uma destas partes receberá o seu número automaticamente, como neste documento, cada seção está sendo incluída no documento-master desta forma. Se eu quiser eliminar uma destas partes, basta-me transformar o respectivo `\input{ }` em um comentário:

```
% \input{ }
```

e L^AT_EX irá ignorar esta linha. Irá numerar corretamente as partes restantes. Isto lhe permite construir documentos de forma progressiva, se estiver satisfeita com o material que já estiver pronto, publique-o! Posteriormente publique uma nova versão com os adendos que lhe parecerem necessários.

Por exemplo, você pode publicar uma lista de exercícios, já preparar, imediatamente, o gabarito, mas deixar o gabarito fora da publicação e depois tornar a publicar a lista de exercícios agora com o gabarito incluído. Porque isto? Por que é muito mais fácil fazer logo o gabarito, no momento em que você estiver preparando as questões e, até é uma forma de testar as questões antes de

passá-las para as alunas. Mas omita o “\input” do gabarito, antecedendo-o do símbolo de comentário, %, e, depois, basta apagar % e voltar a publicar.

Trabalho organizado que L^AT_EX lhe permite! Ou melhor, L^AT_EX permite-lhe dar uma organização superba ao seu trabalho. Crie um banco de questões e apenas com “\inputs” produza a lista de exercícios do dia! Reutilize as questões para fazer listas de exercícios diferentes no próximo semestre. É isto que chamo de reutilização de textos! L^AT_EX lhe permite fazer isto.

Para criar um livro com capítulos, cada arquivo-capítulo deverá ter no começo o comando:

```
\chapter{O título do capítulo}
```

e depois a corpo do capítulo. Neste caso, como se trata de um livro, você deve escolher o estilo `book` ou `report`.

As bibliotecas

```
\usepackage[utf8]{inputenc}
```

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

vão lhe garantir que apareça a palavra “capítulo” em português.

Para criar um texto dividido em seções, cada parte do texto deve começar com

```
\section{O título da seção}.
```

Se quiser incluir uma subseção dentro de uma seção, use

```
\subsection{O título da subseção}.
```

e você tem direito até `subsubsection` apenas estas últimas não serão numeradas. Neste artigo, o arquivo-master tem quatro “\input” porque eu o planejei com 4 seções. Dentro desta seção tem uma “`subsection`” que foi incluída com um `\input`.

Com mais alguns ajustes e você está em condições de publicar as suas notas de aula em **pdf** e usando uma notação Matemática, se for o caso, de primeira linha. Para Linux existem vários pacotes gráficos para produzir L^AT_EX, como `kile`, `latexila` assim como os há para outros sistemas. Eles criam as facilidades para você use os comandos que produzem os símbolos de Matemática apenas clicando em botões. Depois ele tratam de todo o processo de compilação até a produção do arquivo **pdf** e mesmo enviam para a impressora a seu pedido.

Deixe-me insistir, o uso de um documento-master pode parecer a primeira vista uma complicação inútil, mas deixe-me enumerar algumas vantagens a partir do que acabei de dizer:

- Você pode acrescentar ou tirar uma parte do texto com apenas um símbolo de comentário colado á frente de um `input` no documento master. Apagar todo um capítulo, sem perder as informações que ele contém! Ao se arrepender basta apagar o sinal % no documento-master.
- Você pode alterar a ordem dos capítulos ou seções, apenas alterando as linhas dos `inputs` no documento-master.
- Durante o processo de produção do livro, você pode manter desligado todos os capítulos, exceto aquele com que você estiver trabalhando, basta

apagar ou colocar o símbolo de comentário na frente do `input` que corresponder. Para as que gostam de segredos, você pode fazer dois documentos, um para o público em geral, e outro para o seu grupo dominante, neste último incluindo os capítulos com os segredos que o público não deve conhecer...

- Uma vantagem que você somente vai aquilatar com o tempo, você tem uma visão ampla e completa do seu trabalho apenas olhando para o documento-master. Ele funciona como uma espécie de planejamento geral do trabalho.

Estes 4 itens transformam *o que é usar editores de texto tradicionais* ou passar a usar \LaTeX . Eu sempre começo qualquer trabalho escrevendo o *documento master*.

Eu não posso me conter, tenho contar-lhe um segredo sobre o que acabei de fazer neste documento. Na figura (5) página 7,



Figura 5: Variáveis para garantir as mudanças

você pode ver o que fiz no texto.

Eu havia colocado uma lista com 3 itens enumerando a magnificência do \LaTeX quando me ocorreu acrescentar mais um item. Ao revisar o texto descobri o meu erro: “*Estes três itens transformam...*”. Em vez de corrigir **de três para quatro**, resolvi fazer algo mais astucioso que agora vou usar com frequência. Coloquei um contador dentro da lista e usei um comando que incrementa o contador, e finalmente usei o contador na frase que estava errada. Eu usei os comandos seguintes:

- `\setcounter{equacaoUm}{1}` que coloca o valor 1 na variável `equacaoUm`. As variáveis precisam ser declaradas no preâmbulo, Isto é uma necessidade para que \LaTeX possa calcular a quantidade de memória que o documento vai precisar. Quase todas as linguagens de programação exigem isto: declaração antecipada das variáveis.
- `\stepcounter{equacaoUm}` para incrementar de uma unidade a variável `equacaoUm`. Fiz isto em cada novo item.
- Na frase, fora do sistema de itens, escrevi:

“Estes `\arabic{equacaoUm}` itens transformam...”. O comando `\arabic` faz com que o número guardado na variável seja impresso. Não seria possível apenas escrever `equacaoUm` e ver um número, porque \LaTeX não teria meios para compreender que “`equacaoUm`” é uma variável e não uma palavra de nossa língua. O comando `\arabic` informa ao \LaTeX que deve imprimir o número guardado na variável “`equacaoUm`”.

Estes 3 itens lhe explicam como se pode usar uma variável para depois fazer a quantidade de itens que há numa lista, e obviamente, eu repeti aqui o uso de `equacaoUm`. Você poderia talvez se perguntar onde, em que esquisita situação, você poderia usar isto. Resposta: numa lista de exercícios, para colocar uma frase divertida, “cara aluna, esta lista tem exatamente 13 itens. E esta frase pode ficar definitivamente em todas suas listas e nunca estará errada! É o poder do \LaTeX que você nunca encontrará em nenhum outro editor de textos. Obviamente, eu usei aqui, novamente `\arabic{equacaoUm}` tendo antes somado 10 ao valor que nela estava guardado, usando:

```
\addtocounter{equacaoUm}{10}...
```

3.1 Lista de exercícios

Vou terminar esta seção mostrando-lhe o exemplo de *listas de exercícios* cujo resultado você pode consultar nas páginas das disciplinas da *Sobralmatematica*, [7], por exemplo na de Cálculo Multivariado.

O documento que vou apresentar agora não é o arquivo master. Eu uso o arquivo *rost*, que aparece na figura (3), página 15, no qual faço um “input” com um nome adequado sendo o conteúdo semelhante ao que agora passo a apresentar. Isto lhe mostra um aspecto: um documento-master gerencia subdocumentos e estes podem gerenciar seus próprios subdocumentos. Alias, um livro pode ser o resultado final de vários relatórios que foram produzidos com sucesso! Basta encaderná-los em conjunto com um novo *documento-super-master*!

Poderia ser diferente, tudo incluso num único arquivo, mas isto iria contra a minha metodologia de programar \LaTeX ou de escrever programas em qualquer outra linguagem de programação. A ideia básica é sempre trabalhar com módulos específicos. Por exemplo, eu preparo uma lista usando pequenos arquivos que contém questões isoladas de um banco de questões. No banco estão as questões e suas soluções com nomes sugestivos.

Esquemáticamente, a solução consiste dos seguintes passos:

- copiar o esqueleto para um arquivo que se chama `exer_discip_XX.tex` em que `discip` é a raiz que nunca muda de uma disciplina, e `XX` é o número de ordem da lista de exercícios que será aplicada. Recentemente estou usando o ano do calendário como índice.
- Uma certa quantidade de `inputs` com os nomes dos arquivos de exercícios. Esta quantidade corresponde a quantos exercícios que vou querer colocar na lista.

Você viu acima um pequeno exemplo de *documento-super-master*, `rosto.tex`, nele há um `input` para `exer_discip_XX.tex` e neste, há diversos `inputs` para os distintos arquivos de exercícios.

Num *passe de mágica* eu posso montar um livro somente de exercícios!

O arquivo *esqueleto* é o que se encontra na figura (6) página 17. A palavra “esqueleto” é muito usada em computação querendo dizer um molde que você deve completar para obter alguma coisa que funcione. É o caso do arquivo que aparece na figura (6).

Não importa qual seja a disciplina, o *esqueleto* é o mesmo. Ele contém variáveis que vou agora explicar, e, é o uso delas que vai criar as diferenças.

Para simplificar o meu trabalho vou me referir aos comandos do \LaTeX sem sem o uso da contrabarra. Consultando a figura (6) você pode completar a ideia.

- `newcounter` é o comando do \LaTeX que cria uma variável de tipo numérica, inteira: um contador! O comando `setcounter` tem dois parâmetros, o nome do `contador` e o valor que ele vai receber. Em `Pascal` isto seria equivalente a

```
lista := valor;
```

No esqueleto você vê `{ { ** }` que é uma expressão sintaticamente errada, porque tem uma chave que não está sendo equilibrada. Se eu me esquecer de completar este dado o \LaTeX vai parar mencionando o erro. Isto me obriga a preencher todos os dados da nova lista.

- `newcommand` é o comando do \LaTeX para criar um novo comando. Estou criando cinco novos comandos:

Disciplina, Assunto, Objetivo, Chaves e Pagina.

Observe que \LaTeX é sensível á caixa alta, caixa baixa. Quer dizer que `pagina` e `Pagina` seriam comandos diferentes.

Depois de criados, estes comandos serão logo usados, mais abaixo.

- O comando `nocite` seleciona um item do meu banco de bibliografias para incluir neste trabalho. Vou apresentar-lhe, na última seção, o assunto *bibliografia em \LaTeX* , e você pode se adiantar e ler a última seção em conjunto com esta.
- Os dois comandos, `Chaves`, `Pagina` merecem um pouco mais de sua atenção. Eles usam um campo opcional em que há um número inteiro positivo. Indica qual é a quantidade de parâmetros que poderão vir depois. `Chaves` admite três parâmetros, serão três palavras-chave. `Pagina` admite um único parâmetro. Mais abaixo, quando estes comandos forem chamados, está programado um erro para evitar que eles fiquem sem receber os dados. *En passant* podemos perder o preconceito contra *erros*, algumas vezes é importantíssimo *programar erros* em nossos *planos*...
- O par de comandos `begin`, `end` criam ambientes, bem parecido com se faz nos programas em `Pascal`. Nos programas em `C++` ou em `Java` isto

se faz com abertura e fecho de chaves. Aqui estou abrindo um ambiente centralizado, `center`. Verifique que o ambiente foi aberto e fechado.

- Depois foi aberto (e fechado) um ambiente de tabulação. Observe que este ambiente está contido no ambiente centralizador. Este ambiente `tabular` é um dos tipos de tabelas do \LaTeX . A sintaxe das tabelas inclui os seguintes campos:
 - Logo depois de `tabular` um parâmetro que pode conter as seguintes letras
`l,r,c` para `left` (esquerda), `right` (direita), `center` (centro) gerenciando a posição dos dados nos campos da tabela. Neste exemplo tem `ll`, significando que a tabela terá duas colunas em cada linha, e que os dados devem ser puxados para a esquerda. Você pode selecionar qualquer quantidade de campos e com qualquer combinação das três letras. Há mais coisa que pode ser feito aqui que você aprenderá com o tempo. Compare com um exemplo que você pode ver na *Sobralmatematica*, [7, multivariado].
 Pode haver pequenas diferenças, eu estou sempre tentando melhorar o layout das listas.
 - Quem separa os campos, em cada linha, é o símbolo `&`, porém apenas do segundo em diante. Repetindo: o primeiro campo da cada linha não tem o símbolo `&`. Se a tabela tiver apenas uma coluna, não será usado este símbolo de demarcação do único campo.
 - Cada linha é terminada com dupla contra barra: `\\`. O comando `hline` é opcional, serve para traçar uma linha horizontal abaixo dos itens da linha. Você pode usar quantos `hline` desejar, mas cuidado para não poluir o ambiente.
 - O comando `huge` aumenta o tamanho da letra, ele está dentro de duas chaves portanto vale apenas para a palavra `alun@`. Há seis tamanhos para letras:
`Huge, huge, Large, large, small, tiny`
 dispostos em ordem de grandeza do maior para o menor.
 - O comando `today` imprime a data em que o documento for compilado. Devido a pacote `babel` será usado o formato `abnt` para datas.
- O comando `newpage` provoca a mudança de página.
- O ambiente `exercicios`, aberto e fechado mais abaixo, vai imprimir a etiqueta **Exercícios:** e vai forçar que os itens que ele contém sejam impressos com alguma tabulação.
- O comando `noindent` evita que o \LaTeX use tabulação de começo de linha quando encontra uma linha em branco. Como eu usei linhas em branco para aumentar a clareza do texto, é preciso usar o `noindent` para evitar formato de novo parágrafo.

- O ambiente `enumera` vai dar a cada `item` da lista de exercícios um número em sequência. O ambiente `enumera` tem uma variável que pode ser usada para fazer referência ao número de elementos da lista, em lugar daquela “confusão” que eu criei anteriormente, mas o divertido é poder falar do números de elementos de uma lista quando o número dos elementos das listas não aparecem... a mágica do \LaTeX .

O documento termina com dois comandos de bibliografia.

- `bibliographystyle{unsrt}` que recebe dois possíveis parâmetros: `unsrt`, `plain`, para determinar o layout da bibliografia.
- `bibliography{ nome do banco de dados bibliográfico }{ }`

Este segundo item do processamento da bibliografia merece uma atenção especial. Como eu já disse anteriormente, \LaTeX cresce com o apoio dos entusiastas que o utilizam, por que é um sistema de *código aberto*. Algumas pessoas tiveram a ideia de criar um sistema para tratar da bibliografia e nasceu o `BiBTeX`. Ele funciona em cima de um *banco de dados* bibliográficos do autor (ou de uma equipe de autores). O autor, ou a equipe, cria o banco que é um arquivo contendo as informações que aparecem numa bibliografia com todos os autores, artigos, livros, páginas na Internet, que a equipe costuma consultar. Desta forma, quando alguém da equipe for escrever um documento, basta relacionar no início, com o comando `nocite{item}` a lista dos itens necessários ao trabalho que estiver sendo produzido. Dentro do texto o comando `cite{item}` cria a base de dados da citação específica do trabalho em questão.

Eu sempre aconselho aos meus alunos a fazerem experiências com os programas que lhes ofereço, com a observação: *difícilmente destruirão o computador, rodando um programa errado*. Sugiro que você faça experimentos com este arquivo. Faça pequenas mudanças, e experimente. Se der errado você pode retomar o arquivo original e começar novamente. Para isto, sempre faça uma cópia do arquivo, trocando o nome, antes de começar a fazer experiências. Dificilmente você estragará o computador, ou o \LaTeX .

Uma coisa é ler um artigo como este, outra coisa é começar a usar. O meu objetivo aqui não ensinar \LaTeX a ninguém, mas sim, despertar a curiosidade para que as professoras interessadas comecem a usar, como eu fiz, e assim aprendam *a coisa* pelo uso da mesma.

4

Esta é a seção final em que vou selecionar alguns tópicos que seriam usados num trabalho mais avançado. Continua sendo verdade que não espero ensinar nada a ninguém com um texto tão resumido. Observe que o *TeX-book* deve ter mais de 500 páginas e depois dele, muita coisa foi feita inclusive o próprio \LaTeX . Se fosse escutar alguém dizendo que sabe tudo sobre \LaTeX , afaste-se desta pessoa o mais rápido possível. Eu sei o suficiente para escrever um

livro sobre outro assunto, e vou aprendendo mais alguma coisa na medida da necessidade. Então comece a usar e aprenda usando.

4.1 Índice remissivo alfabético

\LaTeX é capaz de montar um índice remissivo alfabético para um texto qualquer, um relatório técnico ou para um livro. Para isto você precisa de executar as seguintes ações:

- Dentro do texto, perto de cada palavra ou assunto que você desejar indexar, use o comando `index{item}`. Quando eu escrevo qualquer texto vou logo indexando tudo que considero importante.
- No documento-master coloque o comando `makeindex` no preâmbulo do documento, antes de `documentclass`. Quando você executar `latex` no documento, será criado um arquivo com a extensão `idx` contendo todas as chaves que você tiver selecionado. Inclusive você pode editar este documento para alterar algum aspecto que lhe interesse.

- Depois de executar `latex documento.tex` execute:

```
makeindex documento.idx
```

é neste momento que será criado o arquivo `documento.ind`. `makeindex` ordena as chaves alfabeticamente e separa os blocos por letra do alfabeto.

- Inclua `documento.ind` na parte final do `documento.tex` e volte a rodar o `latex` para que ele anexe o arquivo de índices ao resultado final. Observe que estou com frequência dizendo que você deve rodar o `latex` mas há outro programa mais recente que produz imediatamente o resultado final em `pdf` que é `pdflatex`. Onde eu disse “rodar \LaTeX ” você pode substituir por “rodar `pdflatex`”.

Artigos não costumam ter índice remissivo, este tem, para lhe mostrar qual é o resultado que se pode esperar deste processo acima descrito.

4.2 Criando novos comandos

Como toda linguagem de programação, \LaTeX admite crescer na direção que interessa ao usuário: pela criação de novos comandos. A figura (7) página 13,

apresenta um resumo do meu arquivo `defestru.sty` em que você pode ver como definir novos comandos. Por exemplo, para não usar o comando `LaTeX` para imprimir \LaTeX eu criei o meu comando é simplesmente

```
\def\latex{\LaTeX}
```

e se desejar uma cópia do meu `defestru.sty` para que você não comece do zero, é só solicitar-me que envio. Mas é bem provável que você encontre coisa melhor procurando na rede. Na figura (7) você tem exemplos de

```
def, newtheorem, newenvironment.
```

No esqueleto da lista de exercícios você tem um exemplo de `newenvironment` que aceita parâmetros, um ou mais parâmetros.

Como usar o seu novíssimo conjunto de definições? **Resposta:**

exatamente como qualquer outro pacote, inserindo no preâmbulo com `usepackage`.

```
%%\ProvidesPackage{defestru}
\def\latex{\LaTeX\ }
\def\cola{\noindent} \def\infinity{\infty}
\def\infinito{\infty}

\def\hoje{Sobral, \number\day \space de \number\month\space
\ifcase\month\or
janeiro\or fevereiro\or mar\cedi o\or abril\or maio\or junho\or
julho\or agosto\or setembro\or outubro\or novembro\or dezembro\fi \space de \number\year}
\def\qed{\framebox{\bf q.e.d.}}
\def\linux{\tt Linux}
\def\Real{R,\$}

\newtheorem{exercicio}{\bf Exerc\'}{\i}
\newtheorem{lab}{\bf Laborat\'}{\i}
\newtheorem{laboratorio}{\bf Laborat\'}{\i}
\newtheorem{exercicios}{\bf Exerc\'}{\i}

\newenvironment{dem}
{\framebox{\bf Dem}: \footnotesize }
{\qed}
\newenvironment{resumo}
{\framebox{\bf Resumo}: \footnotesize }{}
```

Figura 7: Criando um novo pacote

```

\documentclass{report}
\usepackage[utf8]{inputenc}
\usepackage[brazil]{babel}

\begin{document}

\LaTeX \ é um programa para compilar textos.

No início, quem começa a aprender,
tem uma sensação de que está se metendo numa combuca! Tem que usar comandos
para obter efeitos. Passado o primeiro susto descobre que é um compilador de
textos com um poderio fora do comum, somente para dar duas ideias:

\begin{enumerate} \item há
editoras de grande porte como \emph{Academic Press}, um exemplo, que automatizam
todo a sua produção de textos com este programa,

\item é capaz de se comunicar
com linguagens de programação portanto podemos com ele automatizar vários
processos de edição de textos.

\end{enumerate}

Um pequeno exemplo, estou aqui apresentando,
um modelo para produzir documentos padronizados: listas de exercícios.

Depois este modelo pode ser facilmente alterado para produzir
correspondência de um sistema, como do sindicato ou de uma coordenação de
curso.

Se alguém estiver interessado eu ajudo na decolagem, depois que se lançar no
vôo o resto terá de ser consigo.

\end{document}

```

Figura 2: O código fonte

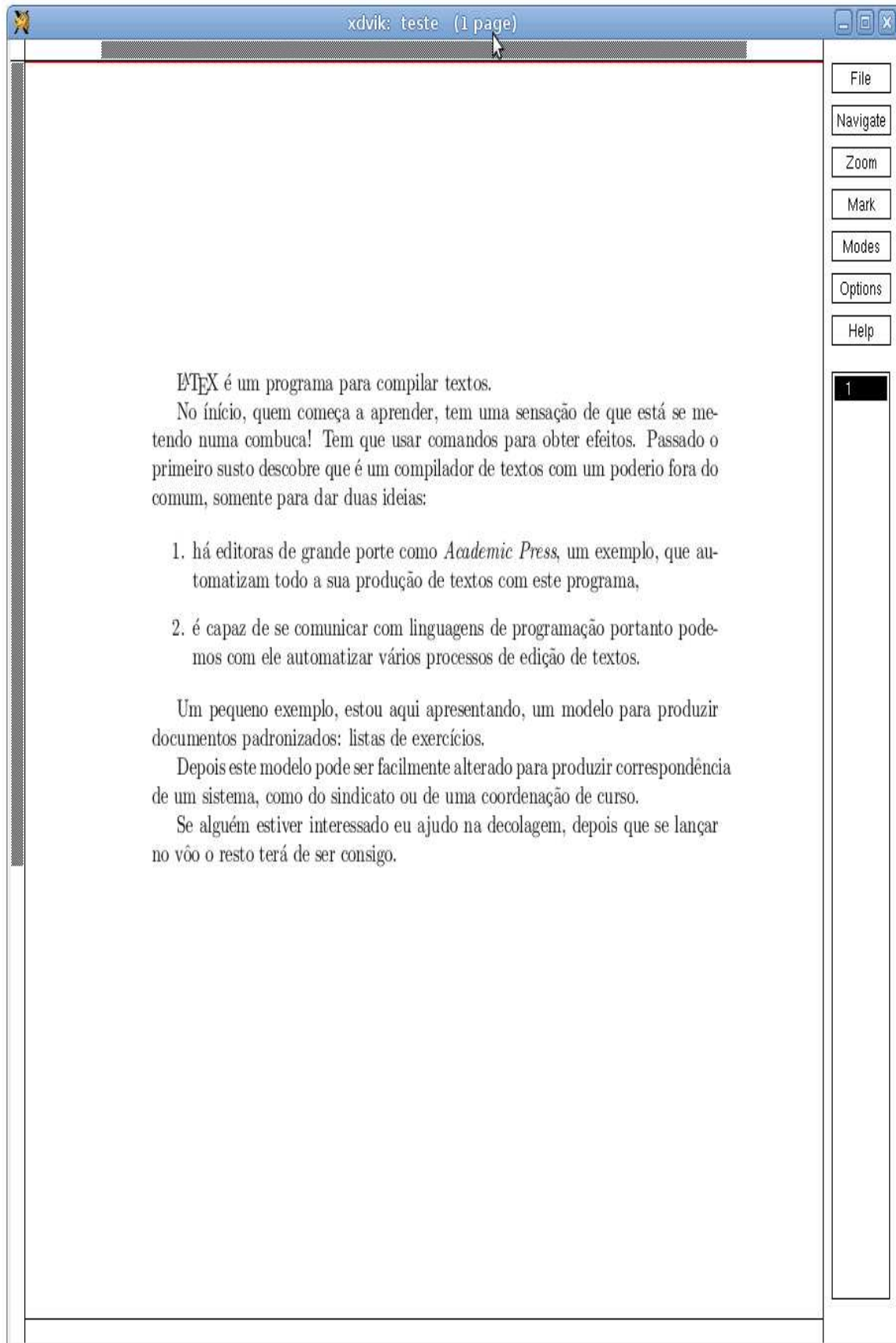


Figura 3: O resultado depois de compilado, em PDF


```

\offset 1cm \hoffset -0.2cm      %% definições da geometria da página e do texto
\textwidth 22cm \textheight 28cm  %% definições da geometria da página e do texto
\headheight 20mm                %% definições da geometria da página e do texto
\oddsidemargin 2.0mm            %% definições da geometria da página e do texto
\evensidemargin 2.0mm          %% definições da geometria da página e do texto
\topmargin -40mm                %% definições da geometria da página e do texto

\documentclass[rm,amstex,a4,12pt,abnt]{article}
\usepackage{amsmath} %% símbolos de Matemática definidos pela AMS
\usepackage[utf8]{inputenc} %% codificação do português
\usepackage[brazil]{babel} %% hifenação portuguesa
\usepackage{epsfig} %% um pacote para trabalhar com gráficos
\usepackage{floatflt} %% um pacote para inserir figuras
\usepackage{color} %% permite colorir letras

\newcounter{equacaoUm} %% definição de uma variável numérica
\newcounter{equacaoDois} %% definição de uma variável numérica
\newcounter{equacaoTres} %% definição de uma variável numérica
\newcounter{equacaoQuatro} %% definição de uma variável numérica
\newcounter{equacaoCinco} %% definição de uma variável numérica

\begin{document}

%% aqui vem as inclusões de textos

\end{document}

```

Figura 4: Documento-master rosto.tex

Índice Remissivo

- índice remissivo, 12
- L^AT_EX, 1
 - instalação, 4
 - kile, LaTeXila, 6
 - pacotes gráficos, 6
 - quem usa, 2
 - redefinindo, 1
- arabic, 8
- babel, 10
- begin, 10
- bibliografia, 11
- BiBTeX, 11
- código fonte, 2
- campos
 - marcador, 10
- capítulo, 6
- centralizando, 10
- chaves, 3
- coisa, 11
- comando
 - como é, 3
 - definição, 1
 - parâmetro, 9
- compilando, 4
- Computer Programming
 - The Art of, 1
- contador
 - imprimir, 8
 - inicializar, 7
 - incrementar, 7
- contadores
 - listas, 7
- defestru.sty, 12
- demarcação
 - campos, 10
- desafio, um dolar, 2
- document
 - begin, 4
- documento
 - master
 - vantagens, 6
 - super-master, 8, 9
 - documento complexo, 2, 16
 - documento simples, 2
- end, 10
- ênfatizar, 4
- enumera, 11
- erro
 - usando
 - para acertar, 9
- erros
 - para lembrar, 9
- esqueleto, 17
- exercícios
 - lista, 8
- experimentar, 11
- figura
 - código fonte, 14
 - documento master, 16
 - inserção, 1
 - inserindo, 16
 - lista de exercícios, 17
 - novo pacote, 13
 - pdf do código fonte, 15
 - variáveis, 7
- fim de linha, 10
- fonte
 - tamanho, 10
- hline, 10
- Huge, 10
- huge, 10
- idx, 12
- imprimir
 - variável, 8
- ind, 12
- index, 12
- input, 5, 9
- inserir

- espaço, 4
- linha, 4
- item, 11
- jogo da velha, 3
- Knuth, Donald, 1
- Large, 10
- large, 10
- LinuX, 4
- lista
 - enumerada, 4
 - exercícios, 8
- listas
 - contadores, 7
- mágica
 - passé de, 9
- makeindex, 12
- marcação
 - linguagem, 2
- markup language, 2
- master
 - documento, 6, 7
- mestre
 - arquivo, 16
 - documento, 16
- não indentação
 - noindent, 10
- newcommand, 9
- newcounter, 9
- newpage, 10
- nocite, 9
- noindent, 10
- pacotes gráficos, 6
- parágrafo, 4
- parametros
 - tabular, 10
- pdflatex, 12
- percentual, 3
- plano do trabalho, 2
- referencias, 5
- seção, 6
- setcounter, 7
- somando
 - com parâmetro, 9
- stepcounter, 7
- subseção, 6
- subsubseção, 6
- tabulação, 10
- tabular, 10
- TeX, 1
- tiny, 10
- today, 10
- usepackage, 4, 13
- variáveis
 - em listas, 7
 - vendo o resultado, 4

Referências

- [1] Piet van Oostrum. Page layout in L^AT_EX . *Dept. of Computer Science; Utrecht University* // http://www.ntg.nl/maps/pdf/16_29.pdf, page 26, 2004.
- [2] Mats Dahlgren. Welcome to the floatflt package! <http://www.ctan.org/tex-archive/macros/latex/contrib/floatflt>, page 4, 1996.
- [3] Hideo Umeki. The geometry package. <http://www.ctan.org/pkg/geometry>, page 42.
- [4] Leslie Lamport. Latex - the macro package for tex. Technical report, <http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html#latex>, 1993.
- [5] Wikipedia. The art of computer programming. http://en.wikipedia.org/wiki/The_Art_of_Computer_Programming.
- [6] Donald Knuth. *The TeXbook*. ADDISON –WESLEY PUBLISHING COMPANY, 1986.
- [7] Sobral Matemática. Sobral matemática. In *Página de Cálculo Multivariado*. <http://www.sobralmatematica.org>, 2012.