

Um curso de L^AT_EX, T_EXworks e Emacs+eev

(↑ Supresa!!!)

Eduardo Ochs (UFF, Rio das Ostras)
UFF, Niterói - 6,7,8 e 13,14,15/fev 2019
no Laboratório do bloco G do Gragoatá

<http://angg.twu.net/#eev>

<http://angg.twu.net/#clg>

(“CLG” = Curso de L^AT_EX do Gragoatá)

Aula 1

O que é o \LaTeX ?

Visão simplista: uma “markup language” – como HTML.
HTML gera webpages (control-P no browser \rightarrow PDFs).

\LaTeX gera PDFs (e DVIs).

\TeX gera PDFs (e DVIs).

O \LaTeX é implementado em cima do \TeX .

Visão realista: o \TeX é uma linguagem de programação;
o \LaTeX é o \TeX “com bibliotecas extras”.

História: $\text{\TeX}76 \rightarrow \text{\TeX}82 \rightarrow \text{\TeX}Book$ (1984)

O $\text{\TeX}Book$ é tão detalhado e preciso,
e o $\text{\TeX}82$ é tão bug-free, que...

O que é o L^AT_EX? (2)

Visão realista: o T_EX é uma linguagem de programação;
o L^AT_EX é o T_EX “com bibliotecas extras”.

História: T_EX76 → T_EX82 → T_EXBook (1984)

O T_EXBook é tão detalhado e preciso,
e o T_EX82 é tão bug-free,
e eles foram tão revolucionários
e tão usados pela comunidade acadêmica
que se tornaram “standard” pra preparar
artigos e livros...

O que é o L^AT_EX? (3)

...que a **comunidade** começou a criar **extensões** pro T_EX...

Exemplo: o arquivo `meuartigo.tex` (em plain text!)

começa com uma parte que faz o T_EX entrar em modo “two-column”, e depois disso todo o resto dele é texto com um mark-up simples.

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

```
\usepackage{lipsum}
```

```
\begin{document}
```

```
\lipsum % Parágrafos 1 a 7 do "Lorem ipsum"
```

```
\end{document}
```

‘%’ inicia um comentário

Linguagens de programação admitem **comentários**.

Tudo numa linha de T_EX depois de um ‘%’ é ignorado.

No exemplo abaixo o único comando `\documentclass` **ativo** é o que seleciona a “classe” `article` com a opção `twocolumn`.

```
%\documentclass[twocolumn]{book}
 \documentclass[twocolumn]{article}
%\documentclass          {book}
%\documentclass          {article}
\usepackage{lipsum}
\begin{document}
  \lipsum % Parágrafos 1 a 7 do "Lorem ipsum"
\end{document}
```

Como assim, “a comunidade”?

TUG: T_EX users group

Revista: TUGBoat (vamos ver como acessá-la!)

Site: <http://www.tug.org/>

CTAN: Comprehensive T_EX Archive Network

Site: <http://mirror.ctan.org/>

T_EXLive: distribuição (com parte do CTAN)

Debian: pacotes `texlive*`

Pro curso nós instalamos a versão “upstream”:

<https://www.tug.org/texlive/quickinstall.html>

O T_EXLive é MUITO grande

Zilhõõõões (mais de 3000!) pacotes,
incluindo FAQs e tutoriais...

Como é que a gente vai lidar com isso?

Resposta: **Emacs+eev**

O que é o Emacs?

Resposta simplista: um editor de texto.

Resposta melhor: um ambiente Lisp.

A idéia de software livre surgiu do Emacs.

(↑) O Emacs também tem/é uma comunidade.

Na década de 70 criaram um editor de texto chamado TECO, que era extensível por “macros”.

As pessoas começaram a criar extensões incríveis pra ele pra deixá-lo mais poderoso.

As pessoas começaram a **compartilhar extensões**, mas o código de cada extensão era feio, elas eram difíceis de entender e às vezes algumas eram incompatíveis entre si.

O que é o Emacs? (2)

“Nós precisamos de um TECO melhor!”

Solução: pegue a linguagem de programação mais elegante e extensível de todas: **Lisp**, adicione algumas funções básicas de “editor de texto” nela, e faça *todo o resto* em Lisp.

O Emacs é um **ambiente Lisp** que geralmente é usado como editor de texto.

“Lisp is available from anywhere”

Lisp:

```

(* 2 3)          --> 6
      (* 4 5)    --> 20
(+ (* 2 3) (* 4 5)) --> 26
(+      6      20) --> 26

```

Essas expressões entre parênteses —
listas sem vírgulas entre os elementos —
são chamadas de “S-expressions”, ou “sexps”.

O Emacs tem um comando chamado ‘eval-last-sexp’
que executa a expressão em Lisp **antes do cursor**.

“Lisp is available from anywhere” (2)

O Emacs tem um comando chamado ‘eval-last-sexp’ que executa a expressão em Lisp **antes do cursor...**
 ou: a “sexp” antes do “point” — e mostra o resultado na “echo area”.

```
(+ (* 2 3) (* 4 5))
      ^      ^^
      |      | \
      6      20 26
```

Se digitarmos ‘M-x find-file RET ~/2019-CLG/’

Isto equivale a: (find-file "~/2019-CLG/")

O Emacs “visita” o diretório “~/2019-CLG/”.

“Lisp is available from anywhere” (3)

LShort: “The Not So Short Introduction to L^AT_EX₂_ε”,
um tutorial que todo mundo usa.

Isto é uma sexp que abre o LShort
num PDF viewer (o xpdf) **na p.23**:

```
(find-lshortpage (+ 14 9) "\\documentclass")
```

O eev é uma extensão pro Emacs que define
“elisp hyperlinks”... esses “elisp hyperlinks” podem
aparecer em comentários —
por exemplo, depois do ‘%’ num arquivo `.tex`.

“Lisp is available from anywhere” (4)

Se editarmos este arquivo `.tex` **no Emacs** podemos seguir os hyperlinks para páginas do LShort usando `‘eval-last-sexp’`:

```
% (find-lshortpage (+ 14 9) "1.6.1 Document Classes")
% (find-lshortpage (+ 14 9)  "\\documentclass")
% (find-lshortpage (+ 14 10)  "twocolumn")
\documentclass[twocolumn]{article}

% (find-lshortpage (+ 14 7) "1.4 Input File Structure")
% (find-lshorttext (+ 14 7)  "\\begin{document}")
\begin{document}
Hello
\end{document}
```

T_EXworks

O T_EXworks é um ambiente beginner-friendly para T_EX e L^AT_EX.

Nós vamos aprender **um pouquinho** de Emacs+eev... o suficiente pra usar (e criar) links para páginas de alguns PDFs que nos interessam — por exemplo o LShort, o manual do T_EXworks, um FAQ do CTAN, estes slides...

Nós vamos aprender a editar arquivos `.tex` com o T_EXworks e **ao mesmo tempo** com o Emacs+eev — que (por enquanto!) vai servir principalmente para os hiperlinks.