

CURRICULUM VITAE

Dados pessoais:

Nome completo: Eduardo Nahum Ochs
Data de nascimento: 7 de fevereiro de 1971
Nacionalidade: brasileira
Sexo: masculino
Estado civil: solteiro

Telefones: (21)3970-2870 - casa
(21)9554-7914 - celular

E-mail: eduardoochs@gmail.com
Home page: <http://angg.twu.net/>
Endereço: R. Joaquim Murтинho .../s...
Santa Teresa
Rio de Janeiro, RJ, Brasil
CEP 20241-320

Formação:

Doutorado em Matemática Aplicada pela PUC-Rio, concluído em agosto de 2003; o doutorado incluiu uma estadia-sanduíche de oito meses — de janeiro a agosto de 2002 — na universidade McGill, em Montreal (Canadá).

Mestrado em Matemática Aplicada pela PUC-Rio, concluído em abril de 1999.

Bacharelado em Matemática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), concluído em agosto de 1996.

Do 2º ano do 1º grau até o 3º ano do 2º grau no Instituto Metodista Bennett, Flamengo, Rio de Janeiro.

Línguas estrangeiras:

Fluente em Inglês.

Ótima compreensão de Francês e Espanhol falados e escritos.

Experiência didática:

Professor substituto da UERJ/FEBF (Duque de Caxias/RJ) durante o primeiro e o segundo semestres de 2004. Cursos ministrados para turmas de graduação:

No primeiro semestre (abril a agosto de 2004):

- Fundamentos da Computação (para Licenciatura em Matemática)
- Geometria Analítica (idem)
- Estatística aplicada à Pesquisa Educacional (para Pedagogia)

No segundo semestre (setembro a dezembro de 2004):

- Fundamentos da Matemática (para Pedagogia)
- Tendências Atuais do Ensino de Ciências (para o curso de Formação de Professores do Ensino Fundamental)

Outros cursos ministrados:

Minicurso de 12 horas sobre programação na linguagem Lua na 1ª Semana de Computação da UFPR (Curitiba), outubro de 2005.

Minicurso de 12 horas sobre programação em Lua no evento “HackerUnion 2”, organizado pela GNU Solutions, setembro de 2004. Local: Faculdades ESEEI, Curitiba.

Apresentações e publicações:

“Simplifying Proofs in Categorical Semantics Using Downcased Types” – apresentação no XIV Encontro Brasileiro de Lógica, ocorrido entre 24 e 28 de abril de 2006, em Itatiata (RJ).

“Emacs e eev, ou: como automatizar quase tudo” – palestra apresentada na 3ª Semana de Software Livre da UNI-Rio, em 16 de outubro de 2005.

“Emacs e eev, ou: como automatizar quase tudo” e “Como ler as entranhas do Debian” – palestras apresentadas na 1ª Semana de Computação da Universidade Federal do Paraná, de 4 a 7 de outubro de 2005.

“A linguagem de programação Lua” (palestra) e “Emacs e eev, ou: como automatizar quase tudo” (workshop), apresentados no “2º Dia D”, evento organizado pelo grupo Debian-RJ, em 13 de agosto de 2005.

“Debian e Software Livre” – Apresentação no “Dia D”, evento organizado pelo grupo Debian-RJ, em 21 de agosto de 2004.

“O que é o esqueleto de uma demonstração?” – Tese de Doutorado, agosto de 2003.

Apresentações feitas durante a bolsa-sanduíche, todas com o título “A System of Natural Deduction for Categories”:

- No encontro “CMS Summer Meeting 2002”, em 17 de junho de 2002.
- No encontro “FMCS 2002”, em 8 de junho de 2002.
- No seminário de Lógica da Universidade de Ottawa, em fevereiro de 2002.
- No seminário de Categorias da Universidade McGill, em janeiro de 2002.

“A system of natural deduction for categories” – apresentado no encontro “Natural Deduction Rio 2001”, ocorrido entre 2 e 6 de julho de 2001 na PUC-Rio.

“‘Set^C is a topos’ has a syntactical proof” – apresentação no seminário de Lógica e Categorias do Centro de Lógica e Epistemologia da UNICAMP, em 25 de abril de 2001.

“Análise não-standard com filtros” – apresentação no Encontro de Lógica da UFF, em 24 de fevereiro de 2000.

“Categorias, filtros e infinitesimais naturais” – Tese de Mestrado, abril de 1999.

Experiência com computadores:

Mais de dez anos de experiência com criação e implementação de domain-specific languages (ou: "mini-linguagens"). As mais interessantes estão descritas nesta página:

<http://angg.twu.net/littlelangs.html>

Fluente nas seguintes linguagens de programação: Emacs Lisp, Lua, shell scripts (Zsh/Bash), Tcl/Tk/Expect, Icon, C, AWK, T_EX e L^AT_EX.

Ótimo conhecimento das seguintes linguagens, sistemas, protocolos e ferramentas: HTML básico, Debian, makefiles, T_EXinfo, Forth, GDB, CGIs. Conhecimentos razoáveis de PHP e Perl. Noções de Python, Ruby, SmallTalk, SQL e PostScript.

Autor do pacote 'eev' para o editor de texto Emacs; o Emacs é um dos dois editores de texto considerados principais em sistemas *NIX. Uma resenha sobre o eev foi publicada no número 13 da revista eletrônica "Brave Gnu World", em 1999.

Alguma experiência com MS-DOS e Microsoft Windows.

Usuário de GNU/Linux desde 1994.

Trabalhos como programador profissional:

De junho/2006 a agosto/2006: implementação de um "daemon" – um analisador de logs de mensagens SMS de celular – em C, usando SWIG para escrever parte das funções do programa em Lua; implementação de interfaces amigáveis para scripts em shell. Empresa: DBA; cliente: Vivo. Rio de Janeiro.

De novembro/2005 a março/2006: manutenção de um sistema legado implementado em C++, SQL e ksh e que roda em servidores AIX. Obs: como eu só tinha acesso ao sistema através de máquinas Windows tive que adaptar o Eev para rodar em Windows para poder automatizar as tarefas necessárias e manter logs do que era feito. Empresa: DBA; cliente: Vivo. Rio de Janeiro.

De dezembro/2004 ao fevereiro/2005: programação de uma jukebox em Linux (sistema embarcado, controlado por apenas 6 botões). Cliente particular, Rio de Janeiro.

De setembro/2000 a novembro/2000: programação de CGIs em PHP. Empresa: Canvas Webhouse, Rio de Janeiro.

De junho/1998 a janeiro/1999: programação de CGIs em Perl. Empresa: Bowne Internet, Rio de Janeiro.

Principais projetos de Software Livre:

- Eev – “uma ferramenta para automatizar quase tudo”, implementada como uma extensão em Emacs Lisp para o editor de texto GNU Emacs, com ferramentas auxiliares escritas em Tcl/Expect, Shell e Awk. Uma resenha sobre o eev foi publicada no número 13 da revista eletrônica “Brave GNU World”, em 1999. Atualmente (março de 2006) o eev está no meio do processo para se tornar um projeto oficial da GNU; ele já foi aceito pela GNU e os detalhes legais já foram resolvidos, mas ainda faltam algumas padronizações e o aceite final. Links principais:

```

http://angg.twu.net/
http://angg.twu.net/eev-article.html
http://angg.twu.net/eev-current/
http://angg.twu.net/eev-current/README.html

```

- Várias mini-linguagens para geração de diagramas para serem incluídos em textos matemáticos escritos em \LaTeX . Os programas que interpretam estas linguagens de diagramas e geram código \LaTeX tiveram várias implementações, em várias linguagens diferentes: em Icon (1997/98), em Icon e Tcl/Tk (1998–2002), e em Lua (2003–). A versão atual processa arquivos em \LaTeX , interpreta certos blocos de comentários como sendo representações ASCII bidimensionais de diagramas categóricos e de árvores de dedução, e produz o código \TeX correspondente. Link principal:

```

http://angg.twu.net/dednat4.html

```

- BlogMe: uma mini-linguagem para geração de código HTML, baseada numa sintaxe em que só colchetes e espaços são caracteres especiais, e com regras de avaliação e expansão de expressões inspiradas em Forth e Lisp; o núcleo do programa consiste em cerca de 100 linhas de código Lua. As páginas principais em HTML do site <http://angg.twu.net/> (cerca de 40) são geradas usando o BlogMe; as outras — cerca de 700 — são geradas usando um antecessor do BlogMe, que é escrito em Tcl e existe desde 1999. Link principal:

```

http://angg.twu.net/blogme.html

```

- Uma mini-distribuição de Debian voltada para pessoas que querem entender como o sistema funciona e fazer as próprias mini-distribuições. Este projeto está em andamento: o sistema já gera um CD que é ao mesmo tempo um “livecd” e um CD de instalação, e ambos funcionam — mas seria de se esperar que a partir de um sistema instalado por este CD seria possível produzir uma versão modificada do próprio CD, e por enquanto o sistema sempre acaba precisando de pacotes extras que ainda não estão sendo incluídos no CD. Observação: os scripts deste projeto que já estão funcionais estão sendo distribuídas junto com o eev, como exemplos de “e-scripts” (veja a descrição do eev).

Principais projetos de Software Livre (continuação):

- “Internet Skills For Disconnected People” — uma variação do projeto anterior em que a mini-distribuição roda dentro de um emulador dentro do sistema que a gerou, e os dois sistemas, o original e o emulado, se comportam como duas máquinas independentes que se comunicam através de uma interface de rede. O objetivo deste projeto é fazer com que pessoas que usam GNU/Linux e que não têm acesso nem a um segundo computador nem a administradores de sistemas possam aprender TCP/IP e redes sozinhas, na prática, usando só o computador que elas têm em casa.

Uma descrição mais detalhada do projeto eev

O eev é uma biblioteca para o Emacs que faz com que o Emacs suporte “e-scripts”. E-scripts são arquivos de “anotações executáveis” sobre como fazer determinadas tarefas no computador.

Um e-script é tipicamente um arquivo de texto no qual alguns pedacos estão em “linguagens humanas”, como Português ou Inglês, e outros pedaços são trechos de código em várias linguagens de computador, ou sequências de comandos para vários programas.

Muitos dos programas textuais interativos, por exemplo shells e linguagens de programação, contêm pequenos editores de texto embutidos — chamados “editores de linha” — que mantêm um histórico dos últimos comandos recebidos e permitem que o usuário edite estes comandos e os execute de novo, com ou sem alterações. A idéia do eev é não precisarmos nos restringir somente a estas ferramentas simplificadas de edição de comandos — com o eev o Emacs passa a poder ser utilizado como um “editor universal de comandos”. Os e-scripts são como históricos globais de comandos dados para o sistema, mas que podem ser editados como arquivos de texto comuns, podem conter todo tipo de comentário e de hiperlink, e podem conter comandos para vários programas e linguagens diferentes.

Não existe nenhum modo de executar um e-script inteiro de uma vez só: cabe sempre ao usuário selecionar trechos de um e-script interativamente e decidir o que fazer com eles. O usuário seleciona a parte a executar (pode ser a linha em que o cursor está, a “região” selecionada pelo Emacs, tudo ao redor do cursor até a primeira ocorrência para a frente e para trás de certos delimitadores, etc) e aí pede que esse trecho seja “executado” de algum modo: interpretado como código Lisp e executado imediatamente pelo Emacs, ou gravado num script temporário que depois vai ser lido por um shell, ou mandado imediatamente para um programa interativo com qual o Emacs estabeleceu um canal de comunicação, etc.

Os “hiperlinks” em e-scripts são simplesmente blocos executáveis, em geral ocupando uma linha só ou parte de uma linha, que quando são executados têm uma ação correspondente à de seguir um hiperlink. Um exemplo:

```
# (find-node "(make)Automatic" "$^")
```

quando o Emacs executa essa expressão entre parênteses (essa expressão é um comando em Emacs Lisp) ele abre a página intitulada “Automatic” do manual do GNU Make e leva o cursor para logo depois da primeira ocorrência do string “\$^” no texto.

Repare que este tipo de hiperlink pode aparecer no meio de blocos de código pra praticamente qualquer programa ou linguagem — já que quase todos os programas num sistema *NIX aceitam “comentários”. Por exemplo, em shell scripts todas as linhas começadas com “#” são tratadas como comentários e ignoradas.