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 Murtinho 786/s302
	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:

"Boostrapping a Forth in 40 lines of Lua code" – artigo aceito para o livro "Lua Programming Gems". O livro está em fase de edição e deve ser publicado em 2008.

"Symplifying 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.

" '$\mathbf{Set}^{\mathbf{C}}$ 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.

Apresentações recentes em seminários:

Todas as apresentações abaixo foram feitas no Seminário de Lógica da UFF e correspondem a pesquisa que ainda não foi publicada formalmente.

- 13/03/2008: Feixes para Não-Categoristas
- 28/11/2006: Sistemas de Tipos com Dicionários - Parte IV: Uma Aplicação
- 14/11/2006: Sistemas de Tipos com Dicionários - Parte III
- 24/10/2006: Sistemas de Tipos com Dicionários - Parte II
- 03/10/2006: Sistemas de Tipos com Dicionários
- 06/12/2005: A Language for Categorical Semantics: System DNC (III)
- 01/11/2005: A Language for Categorical Semantics: System DNC (II)
- 20/10/2005: A Language for Categorical Semantics: System DNC (I)

Para mais informações veja a página do seminário:

`http://www.uff.br/grupodelogica/`

Participação em mesas-redondas:

"Software Livre, TV Digital e Rádio Digital: desafios e perspectivas para a Educação" — em 13/nov/2007, no 5º Encontro de Educação e Tecnologia de Informação e Comunicação, Universidade Estácio de Sá, Rio de Janeiro.

Texto da apresentação e detalhes disponíveis em:

`http://angg.twu.net/2007mesa.html`

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, TeX e LaTeX.

Ótimo conhecimento das seguintes linguagens, sistemas, protocolos e ferramentas: HTML básico, Debian, makefiles, TeXinfo, Forth, GDB, CGIs. Bom conhecimento de C++ (com a biblioteca Qt) e Ruby. Conhecimentos razoáveis de PHP e Perl. Noções de Python, 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.

Principais trabalhos como programador profissional:

De dezembro/2007 a fevereiro/2008: implementação da primeira versão da interface homem-máquina de um sistema de rastreio ótico de foguetes. Omnisys Engenharia, São Caetano do Sul, SP.

De julho/2007 a outubro/2007: início da implementação de um debugador gráfico para a linguagem Lua, integrando um toolkit de janelas desenvolvido pelo TecGraf ("IUP"), as rotinas gráficas ("Scintilla") de um editor de texto ("Scite"), e um debugador para Lua com uma interface de linha de comando ("ldb"). Como programador free-lancer, no TecGraf, PUC-Rio, Rio de Janeiro.

De abril/2007 a junho/2007: implementação de novas funções num sistema de backup incremental escrito em Ruby. Digirati Informática, Rio de Janeiro.

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 2008) 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/blogme3.html
```

Principais projetos de Software Livre (continuação):

• Middle-C: uma variação da linguagem C em que os operadores '`[index]`', '`[]`', '`(args)`' e '`.fieldname`' passam a ser "prefix operators" ao invés de "postfix". Com esta modificação na sintaxe e umas poucas outras obtemos uma linguagem na qual todos os tipos do C podem ser expressos canonicamente como strings sem espaços, e as declarações ficam bem mais simples. Por exemplo:

```
            (C)                          (Middle-C)

    int *((*a)[20]);              int*[20]* a;
    int main(int argv,            int(int,char[]*) main;
             char *argv[]);
```

A idéia principal do Middle-C não é ser uma linguagem autônoma, mas implementar todo o sistema de tipos do C em Lua. A partir de um interpretador Lua chamado a partir de um programa em C podemos ler uma biblioteca em Lua ("peek.lua") de poucas centenas de linhas que entende arquivos de declarações do C (".h"s) convertidos para a sintaxe do Middle-C, e aí examinar e manipular as estruturas de dados do programa em C interativamente, de um modo que é tecnicamente bem mais simples do que, por exemplo, pela linha de comando do GDB. Link principal:

`http://angg.twu.net/peek.html`

• MiniForth (descrito no artigo "Boostrapping a Forth in 40 lines of Lua code"). Um sistema Forth experimental implementado sobre Lua que explora vários modos de traduzir as idéias do Forth para um ambiente em que a "memória" não é mais um array de bytes, e sim um array de "células", em que cada uma pode conter um número, um string, uma função, um array, etc, e onde muitas vezes é mais prático usar funções do Lua – por exemplo, para "regexp matching" no parser – do que implementar tudo a partir do zero em Forth "puro", operando diretamente sobre os bytes da memória. Link principal:

`http://angg.twu.net/miniforth-article.html`

• RubyForth – escrito em conjunto com Marc D. Simpson – é uma versão "usável" do MiniForth, que implementa boa parte das funções descritas no ANS Forth Standard e várias funções e estruturas inspiradas por Forths escritos pelo criador da linguagem, que se afastou do mainstream da comunidade no início da década de 1990. Link principal:

`http://angg.twu.net/rubyforth.html`

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.