

Aluno: Pedro G. Branquinho
Orientador: Dr. Wei-Liang Qian,

Softwares Livres na Academia e na Indústria

Lorena, São Paulo
14 de junho de 2021

Resumo

Palavras-chaves: software livre. automação. freqtrade. indústria. academia.

Lista de ilustrações

Figura 1 – Genealogia Distribuições Linux	6
Figura 2 – EXWM - Emacs X Window Manager	7
Figura 3 – Diagrama esquemático simplificado - comunicação cliente-servidor	8
Figura 4 – Diagrama esquemático - comunicação cliente-servidor	8

Sumário

1	INTRODUÇÃO	4
1.1	Objetivo	4
2	REVISÃO BIBLIOGRÁFICA	5
2.1	Open Source	5
2.1.1	Diversidade	5
2.2	O Linux	6
2.2.1	Origem Histórica	6
2.2.2	O Emacs	7
2.3	Aplicações na Indústria	7
2.3.1	Freqtrade	7
2.3.2	OR-Tools	9
2.4	Aplicações Acadêmicas	9
2.4.1	DifferentialEquations.jl	9
2.4.1.1	Portabilidade em Julia	10
2.4.1.2	Portabilidade em Python	10
2.4.1.3	Portabilidade em R	10
2.4.2	O L ^A T _E X	11
2.4.2.1	Classe Canônica ABNT de produção científica	11
2.5	Trabalhos Canônicos na Área de Computação	12
2.5.1	Structure and Interpretation of Classical Mechanics (SCIM)	12
2.5.2	Structure and interpretation of computer programs (SICP)	12
2.5.3	SICMUtils - Portabilidade de (SCIM) em Clojure	13
3	MATERIAIS E MÉTODOS	14
4	RESULTADO E DISCUSSÕES	15
5	CONCLUSÃO	16
	REFERÊNCIAS	17

1 Introdução

Na formação de um engenheiro físico, o qual, por definição, é um profissional generalista, os softwares abertos (FOSS - Free and Open Source Software) e a participação da comunidade Open Source são detrimenais para sua formação.

A diversidade dos quais softwares extensíveis acarretam ([subseção 2.1.1](#)) podem mudar completamente a experiência do usuário, e o trazer mais próximo do papel de desenvolvedor. Essa experiência não necessita de ser exclusiva de cientistas da computação ou profissionais de TI. Pois, a programação pode ser encarada tanto como ciência e arte ([KNUTH, 1968](#)).

Os Softwares Abertos possuem quatro liberdades p treas [se   o 2.1](#), garantindo os direitos de estudo, c pia, modifica  o e redistribui  o.

Bem como a ci ncia se beneficia com seus r pidos avan os, de uma comunidade global de participantes, com as mais distintas especializa  es profissionais. Tamb m, beneficia-se a computa  o com a comunidade aberta, e especializa  o ecl tica, tanto de membros quanto de softwares.

1.1 Objetivo

Demonstramos a defasagem que um profissional de engenharia apresentaria, sem forte forma  o dentro da computa  o. Ademais, ao partir da gama de aplica   es, em estado da arte, as quais s o partilhadas de forma aberta e livre, pretende-se reinterar o caso da necessidade de compreens o do fen meno dos softwares abertos. Pois, essa l gica e din mica n o possui paralelos nem na economia, nem na comunidade cient fica ([HIPPEL; KROGH, 2003](#); [PETERS, 2009](#)).

H  at  debates acirrados sobre o sentido de Open Science, um termo que recentemente se popularizou e o qual constitui claro paralelo com o movimento Open Source. Por m, mal compreendido e, raramente, debatido sob esse prisma, dentro das ci ncias sociais. Cita-se o mais citado dos artigos na busca no Google Scholar “The future(s) of open science” , o qual somente cita tr s vezes o termo Open Source, em tom dismissivo ([MIROWSKI, 2018](#)). Assim, argumenta-se que a forma  o b sica do engenheiro na  rea da computa  o precisa ser s lida, bem como o entendimento das for as que moldaram esse movimento, de forma a poder entender o futuro em que caminhamos, de forma cr tica.

2 Revisão Bibliográfica

2.1 Open Source

Qualquer programa que permita o usuário-programador ter as seguintes liberdades:

1. Direito de rodar o programa, como você desejar, para qualquer fim.
2. Direito ao acesso ao código-fonte, para estudá-lo.
3. Direito de cópia e distribuição.
4. Direito à modificação do software.

De maneira prática, a comunidade Open Source, fundamentalmente, se baseia no compartilhamento de suas configurações. As vantagens de existirem inúmeras outras pessoas utilizando o mesmo software é de que a melhoria da fronteira do programa é expandida de forma acrescida, em comparação a de um time restrito de usuários.

2.1.1 Diversidade

Dado que um direito fundamental dos softwares livres é a modificação e propagação das versões modificadas, existe uma diversidade de expressividade, sem paralelos em outras áreas da tecnologia.

Por exemplo, uma parte de software fundamental na configuração de um computador é seu gerenciador de interfaces (Window Manager). Onde, um programa é devotado a gerenciar como outros programas gráficos devem se dispor na tela de computador.

Enquanto sistemas operacionais (Operational Systems) privados, como Windows e MacOS possuem versões lançadas frequentemente - vinte e cinco versões lançadas de Windows. O Windows possui apenas quatro versões, com suporte ativo ([MICROSOFT..., 2021](#)).

São vinte lançamentos de MacOS, e quatro verões mantidas ([MACOS..., 2021](#)).

Essa estreiteza de versões se dá, dentre os fatores, pois os usuários são cercados do direito de estender ou alterar os comportamentos programados no sistema. Assim, vítimas do suporte descontinuado e de sua atualização de versões restritivas.

Em contra partida, existem, paralelamente, por volta de 278 distribuições de

Linux ([LINUX..., 2021](#)). Onde, existem as distribuições raízes, com princípios e filosofias de desenvolvimentos teóricos e práticos diferentes.

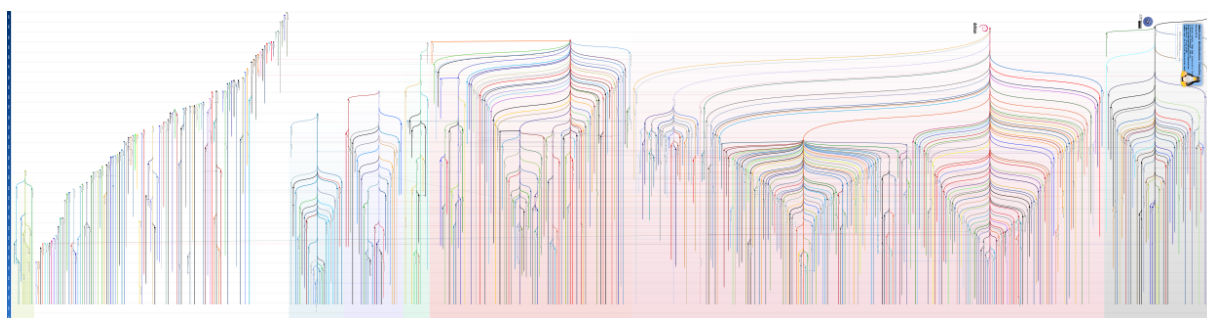
Assim, bem como em qualquer outro escopo de software, a variação dos softwares abertos e livres (FOSS) sempre serão superiores aos monopolizados.

2.2 O Linux

Existem distribuições raízes de linux, das quais muitas distribuições existem como ramificações. Nomeia-se, de forma genérica, devido aos princípios base de uma classe de distribuições, como famílias. Cita-se algumas das mais influentes e populares, Red Hat Linux, Debian, CentOS, Fedora, Pacman-based, OpenSUSE, Gentoo-based, Ubuntu-based, Slackware, Open Sourced-based e as distribuições Independetes.

É possível apreciarmos visualmente a riqueza de distribuições pela [Figura 1](#).

Figura 1 – Genealogia Distribuições Linux



Histórico de evolução das distribuições Linux ([LINUX..., 2021](#))

2.2.1 Origem Histórica

O projeto do GNU/Linux iniciou-se separadamente, por duas frentes. O GNU - abreviação de, GNU's Not Unix - por usuários revoltados com o sistema de segurança dos computadores do MIT (Laboratory of Computer Science - LCS) ([STALLMAN, 2002](#); [EMACS..., 2021](#)). Dentre eles, o ainda ativo Richard Stallman, após já dez anos de evolução do editor de texto ([EMACS..., 2021](#)).

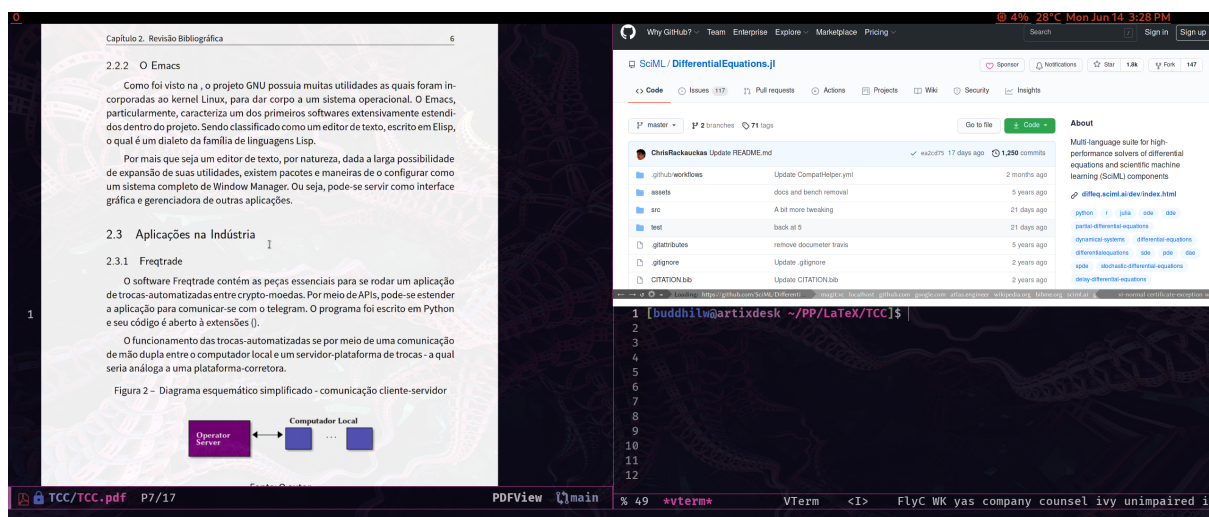
Paralelamente, Linus Torvalds desenvolveu um sistema operacional portátil aberto, como sua tese de mestrado ([TORVALDS, 1997](#)). Por fim, houve uma junção dos projetos, os quais colaboravam o Linux, como sistema operacional, e GNU com todas as aplicações utilitárias do sistema ([STALLMAN, 1997](#)).

2.2.2 O Emacs

Como foi visto na , o projeto GNU possuía muitas utilidades as quais foram incorporadas ao kernel Linux, para dar corpo a um sistema operacional. O Emacs, particularmente, caracteriza um dos primeiros softwares extensivamente estendidos dentro do projeto. Sendo classificado como um editor de texto, escrito em Elisp, o qual é um dialeto da família de linguagens Lisp.

Por mais que seja um editor de texto, por natureza, dada a larga possibilidade de expansão de suas utilidades, existem pacotes e maneiras de o configurar como um sistema completo de Window Manager. Ou seja, pode-se servir como interface gráfica e gerenciadora de outras aplicações.

Figura 2 – EXWM - Emacs X Window Manager



Fonte: foto do ambiente de WM do autor.

A Figura 2 é um exemplo do ambiente de desktop totalmente manuseado por meio do Emacs, utilizando-se do EXWM. Pode-se notar que é possível rodar browsers modernos, bem como renderizadores de imagens e PDF.

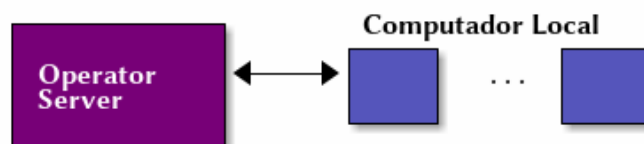
2.3 Aplicações na Indústria

2.3.1 Freqtrade

O software Freqtrade contém as peças essenciais para se rodar um aplicação de trocas-automatizadas entre crypto-moedas. Por meio de APIs, pode-se estender a aplicação para comunicar-se com o telegram. O programa foi escrito em Python e seu código é aberto à extensão, possuindo cento e quarenta e sete contruibuidores, com existência datando de Maio de 2017 (FANG et al., 2020).

O funcionamento das trocas-automatizadas se por meio de uma comunicação de mão dupla entre o computador local e um servidor-plataforma de trocas - a qual seria análoga a uma plataforma-corretora.

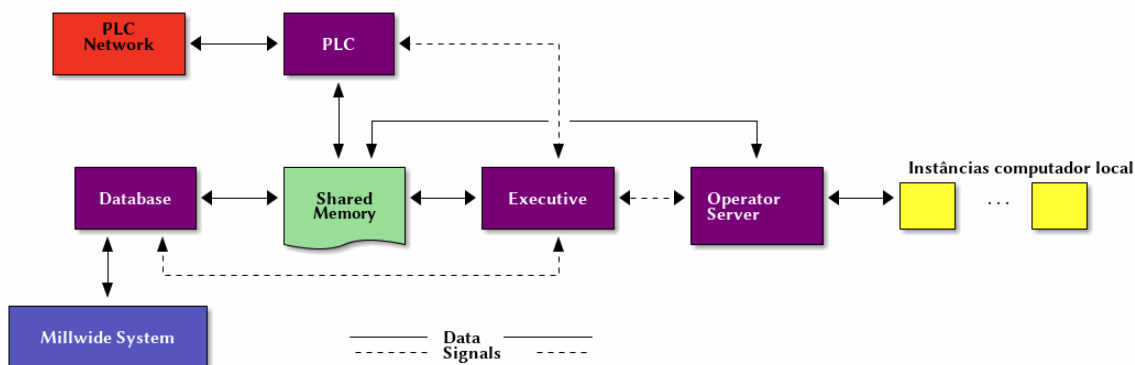
Figura 3 – Diagrama esquemático simplificado - comunicação cliente-servidor



Fonte: O autor.

Grande parte do trabalho de se escrever um robô autômato, para qualquer fim, compraz em programar protocolos de comunicação com servidores. Pois, o robô deverá ser capaz de dizer o servidor quais operações devem ocorrer, tanto gerando respostas ao cliente - dados a serem armazenados no computador local -, quanto operações internas ao servidor - como, executar uma compra e venda de criptomoeda.

Figura 4 – Diagrama esquemático - comunicação cliente-servidor



Fonte: O autor.

Assim, um diagrama completo o qual captura essa complexidade poderia esquematicamente ser visto na [Figura 4](#).

Ao se utilizar o software aberto, toda essa complexidade da [Figura 4](#) se torna mais próxima à [Figura 3](#), pois toda a abstração-estrutural já se encontra escrita. Basta, para quem decidir utilizar o programa ler, entender e modificar o conteúdo disponível na documentação do projeto.

2.3.2 OR-Tools

A ferramenta OR-Tools traduz-se em bibliotecas largamente desenvolvida pela comunidade aberta, e a empresa Google (); utilizável em C++, Python, Java, (Closure), C#, .Net.

OR-Tools possui a utilidade de resolver problemas como agendamento de horários, o qual, estruturalmente, assemelha-se aos algoritmos de solução de problemas como Sudoku ().

De forma geral, esses problemas caem dentro de um ramo da matemática e computação chamado Problemas de Optimização Constrita (Constraint Optimization)

2.4 Aplicações Acadêmicas

2.4.1 DifferentialEquations.jl

A biblioteca numérica DifferentialEquations possui uma das melhores performances de softwares numéricos que existem ([RACKAUCKAS; NIE, 2017b](#)). Sua performance é comparativa à implementações em FORTRAN e C. Existem ports da biblioteca para Python e R, porém, desenvolveu-se em Julia.

Dentre a categoria de problemas possíveis de se resolver, utilizando-se ferramentas próprias do pacote, lista-se as seguintes categorias de equações diferenciais ([RACKAUCKAS; NIE, 2019](#); [RACKAUCKAS; NIE, 2017a](#); [RACKAUCKAS; NIE, 2018](#); [SYKORA et al., 2020](#); [RACKAUCKAS et al., 2018](#); [RACKAUCKAS et al., 2019](#); [RACKAUCKAS et al., 2020](#); [GOWDA et al., 2019](#); [MA et al., 2021](#)),

- Equações discretas (Estocásticos discretos (Gillepie/Markov), e mapas funcionais)
- Equações diferenciais ordinárias (ODEs)
- Equações estocásticas ordinárias (Integrações simpléticas, Método IMEX)
- Equações estocásticas diferenciais algébricas (SODEs or SDEs)
- Equações diferenciais aleatórias (RODEs ou RDE)
- Equações diferenciais algébricas (DAEs)
- Equações diferenciais com retardo (DDEs)
- Equações diferenciais neutras, retardadas e algebricamente retardadas (NDDES, RDDEs e DDAEs)
- Equações diferenciais estocásticas retardadas (SDDEs)
- Suporte parcial de soluções em equações diferenciais estocásticas neutras, retardadas e retardadas algebraicas (SNDDEs, SRDDEs, SDDAEs)
- Equações mistas discretas e contínuas (Jump Diffusions)

- Equações diferenciais parciais (estocásticas) ((S)PDEs) - ambas com portabilidade de métodos finitos e/ou diferenças finitas)

2.4.1.1 Portabilidade em Julia

Para se utilizar do pacote, basta utilizar os seguintes comandos

```
# Carrega o gerenciador de pacotes, Pkg e instala, se preciso, o pacote.
using Pkg
Pkg.add("DifferentialEquations")
# Porta as utilidades do pacote, sem necessitar de referí-lo no meio do código
using DifferentialEquations
```

2.4.1.2 Portabilidade em Python

Num terminal, utilizando-se do gerenciador de pacotes pip,

```
pip install diffeqpy
```

Num terminal, rodando-se o interpretador de Python,

```
>>> import diffeqpy
>>> diffeqpy.install()
```

Por fim, opcionalmente, utilize numba para aumentar a performance do código,

```
pip install numba
```

Ademais, apenas repita o comando, dentro de um arquivo python,

```
import diffeqpy
```

2.4.1.3 Portabilidade em R

Para instalação,

```
install.packages("diffeqr")
```

Na primeira chamada de,

```
diffeqr::diffeq_setup()
```

Será feito o download do DifferentialEquations.jl. Um subconjunto específico do pacote é ativamente mantido [no CRAN](#).

2.4.2 O \LaTeX

O \LaTeX possui separação entre as tarefas de produção de um documento. A linguagem permite-nos separar as tarefas de formatação do texto, da escrita de seu conteúdo. Desta forma, o usuário concentra-se exclusivamente em seu conteúdo, em um estágio da escrita do documento. E, na formatação de sua aparência, em outro momento.

Assim, ganha-se em qualidade de produção. Bem como, ganha total autonomia sob o documento, pois a programação da disposição gráfica dos elementos textuais pode ser programada - isto é, modificada indefinidamente, a partir dos comportamentos padrões dos pacotes utilizados. O sistema tipográfico de \LaTeX chegou a ser considerado o sistema digital de tipografia mais sofisticado que existe, devido a essa paradigma de programação funcional, bottom-up ([HARALAMBOUS, 2007](#)).

O \LaTeX , tecnicamente, é a junção do sistema de tipografia $T_E X$, inventado por Donald Knuth, para tipografia de alto nível ([KNUTH, 1986](#)); com os poderosos macros que facilitam a extensão do programa $T_E X$, a qual damos o nome de \LaTeX . O \LaTeX foi inicialmente desenvolvido por Leslie Lamport, com seus pacotes fundamentais de formatação ([LAMPART, 1994](#)). O \LaTeX , por conseguinte, não é somente uma linguagem de tipografia de alto nível, mas também um conjunto de macros para facilitar a tipografia em si. Qualifica-se, assim, como um sistema de preparação de documentos; uma linguagem markup de domínio específico.

2.4.2.1 Classe Canônica ABNT de produção científica

Documentos sob os requisitos das normas ABNT (Associação Brasileira de Normas Técnicas) para elaboração de documentos técnicos e científicos brasileiros - como artigos científicos, relatórios técnicos, trabalhos acadêmicos, como teses, dissertações, projetos de pesquisa e outros documentos do gênero ([ABNTEX, 2012](#)) - é ao que se chama classe canônica ABNT.

Os documentos indicados tratam-se de “Modelos Canônicos” , ou seja, de modelos que não são específicos a nenhuma universidade ou instituição, mas que implementam exclusivamente os requisitos das normas da ABNT, Associação Brasileira de Normas Técnicas. ([ARAUJO, 2018](#), Cap. 1)

As normas as quais prescrevem o modelo canônico são:

- **ABNT NBR 6022:2018:** Informação e documentação - Artigo em publicação periódica científica - Apresentação.
- **ABNT NBR 6023:2002:** Informação e documentação - Referência - Elaboração.

- **ABNT NBR 6024:2012:** Informação e documentação - Numeração progressiva das seções de um documento - Apresentação.
- **ABNT NBR 6027:2012:** Informação e documentação - Sumário - Apresentação.
- **ABNT NBR 6028:2003:** Informação e documentação - Resumo - Apresentação.
- **ABNT NBR 6029:2006:** Informação e documentação - Livros e folhetos - Apresentação.
- **ABNT NBR 6034:2004:** Informação e documentação - Índice - Apresentação.
- **ABNT NBR 10520:2002:** Informação e documentação - Citações.
- **ABNT NBR 10719:2015:** Informação e documentação - Relatórios técnicos e/ou científico - Apresentação.
- **ABNT NBR 14724:2011:** Informação e documentação - Trabalhos acadêmicos - Apresentação.
- **ABNT NBR 15287:2011:** Informação e documentação - Projeto de pesquisa - Apresentação.

2.5 Trabalhos Canônicos na Área de Computação

2.5.1 Structure and Interpretation of Classical Mechanics (SCIM)

A biblioteca científica de simulações de mecânica clássica, em Scheme (um dialeto de Lisp), foi escrito com intento de ser utilizado em cursos de mestrado no MIT. Acompanhado à biblioteca, existe o livro, o qual serve de material didático ao curso ([SUSSMAN; WISDOM, 2015](#)).

2.5.2 Structure and interpretation of computer programs (SICP)

O curso de Scheme (SICP), o qual é ensinado como matéria básica de computação no MIT, possui como acompanhantes um dos mais influentes livros já escritos na história da computação ([ABELSON; SUSSMAN, 1996](#)). O curso é pioneiro em aprofundar-se epistemologia da programação. Onde, o uso de Scheme, o qual possui notação uniforme para todas as funcionalidades da língua foi revolucionário. Conquanto, em contraste aos outros cursos de fundamentos da computação focavam em ensinar a linguagem mais amplamente utilizada no momento.

O curso SCIM ([subseção 2.5.1](#)) foi um precursor de SICP. O curso continua ativo e o livro continua a ser utilizado amplamente no MIT, e em inúmeras outras faculdades. Uma lista não exaustiva pode ser encontrada em <https://mitpress.mit.edu/sites/default/files/sicp/adopt-list.html>, em que pelo menos, vinte universidades o adotam, em mais de oito países.

2.5.3 SICMUtils - Portabilidade de (SCIM) em Clojure

Existe uma biblioteca reescrita em clojure, a qual porta as funcionalidades descritas na biblioteca científica SCIM ([SMITH; RITCHIE, 2016](#)). Essa biblioteca é escrita numa das mais modernas languages de computação com adoção amplamente da Indústria, Clojure. Recentemente, a maior empresa utilizado de Clojure até então, a Cognitect, a qual inventou a língua foi comparada pelo Nubank ([HAMILTON, 2020](#)).

A língua é tão poderosa que as implementações não são apenas numéricas, nem são somente simbólicas. É possível utilizar-se da mesma função (procedure) para se fazerem cálculos numéricos ou simbólicos, tendo apenas de se mudar os argumentos de entrada (inputs).

3 Materiais e Métodos

4 Resultado e Discussões

5 Conclusão

Referências

ABELSON, H.; SUSSMAN, G. J. Structure and interpretation of computer programs. [S.l.]: The MIT Press, 1996. Citado na página 12.

ABNTEX, E. A classe abntex2: Modelo canônico de trabalhos acadêmicos brasileiros compatível com as normas abnt nbr 14724: 2011, abnt nbr 6024: 2012 e outras.[sl], 2012. <http://code.google.com/p/-abntex2/>. Citado, v. 2, p. 2, 2012. Citado na página 11.

ARAUJO, L. C. A classe abntex2. Documentos técnicos e científicos brasileiros, 2018. Citado na página 11.

EMACS History. 2021. <<https://www.emacswiki.org/emacs/EmacsHistory>>. Citado na página 6.

FANG, F. et al. Cryptocurrency trading: a comprehensive survey. arXiv preprint arXiv:2003.11352, 2020. Citado na página 7.

GOWDA, S. et al. Sparsity programming: Automated sparsity-aware optimizations in differentiable programming. 2019. Citado na página 9.

HAMILTON, A. Brazilian challenger Nubank acquires firm behind Clojure and Datomic. 2020. <<https://www.fintechfutures.com/2020/07/brazilian-challenger-nubank-acquires-firm-behind-clojure-and-datomic/>>. Citado na página 13.

HARALAMBOUS, Y. Fonts & encodings. [S.l.]: ” O’ Reilly Media, Inc.” , 2007. Citado na página 11.

HIPPEL, E. v.; KROGH, G. v. Open source software and the “private-collective” innovation model: Issues for organization science. Organization science, Informs, v. 14, n. 2, p. 209–223, 2003. Citado na página 4.

KNUTH, D. E. The art of computer programming, vol 1: Fundamental. algorithms, p. 187, 1968. Citado na página 4.

KNUTH, D. E. TEX: the Program. [S.l.]: Addison-Wesley, 1986. Citado na página 11.

LAMPORT, L. LATEX: a document preparation system: user’ s guide and reference manual. [S.l.]: Addison-wesley, 1994. Citado na página 11.

LINUX, list of distributions. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/List_of_Linux_distributions>. Citado na página 6.

MA, Y. et al. ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling. 2021. Citado na página 9.

MACOS version history. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/MacOS_version_history>. Citado na página 5.

- MICROSOFT, list of operating systems. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/List_of_Microsoft_operating_systems>. Citado na página 5.
- MIROWSKI, P. The future (s) of open science. *Social studies of science*, SAGE Publications Sage UK: London, England, v. 48, n. 2, p. 171–203, 2018. Citado na página 4.
- PETERS, M. A. Open education and the open science economy. *Yearbook of the National Society for the Study of Education*, v. 108, n. 2, p. 203–225, 2009. Citado na página 4.
- RACKAUCKAS, C. et al. Diffeqflux.jl-a julia library for neural differential equations. arXiv preprint arXiv:1902.02376, 2019. Citado na página 9.
- RACKAUCKAS, C. et al. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. arXiv preprint arXiv:1812.01892, 2018. Citado na página 9.
- RACKAUCKAS, C. et al. Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385, 2020. Citado na página 9.
- RACKAUCKAS, C.; NIE, Q. Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory. *Discrete and continuous dynamical systems. Series B, NIH Public Access*, v. 22, n. 7, p. 2731, 2017. Citado na página 9.
- RACKAUCKAS, C.; NIE, Q. Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software, Ubiquity Press*, v. 5, n. 1, 2017. Citado na página 9.
- RACKAUCKAS, C.; NIE, Q. Stability-Optimized High Order Methods and Stiffness Detection for Pathwise Stiff Stochastic Differential Equations. arXiv:1804.04344 [math], 2018. Disponível em: <<http://arxiv.org/abs/1804.04344>>. Citado na página 9.
- RACKAUCKAS, C.; NIE, Q. Confederated modular differential equation apis for accelerated algorithm development and benchmarking. *Advances in Engineering Software, Elsevier*, v. 132, p. 1–6, 2019. Citado na página 9.
- SMITH, C.; RITCHIE, S. SICMUtils: Functional Computer Algebra in Clojure. 2016. <<http://github.com/sicmutils/sicmutils>>. Citado na página 13.
- STALLMAN, R. Linux and GNU - GNU Project - Free Software Foundation. 1997. <<https://www.gnu.org/gnu/linux-and-gnu.html>>. Citado na página 6.
- STALLMAN, R. My lisp experiences and the development of gnu emacs (transcript of richard stallman’s speech, 28 oct 2002, at the international lisp conference). URL (consulted December 2003): <http://www.gnu.org/gnu/rms-lisp.html>, 2002. Citado na página 6.
- SUSSMAN, G. J.; WISDOM, J. Structure and interpretation of classical mechanics. [S.l.]: The MIT Press, 2015. Citado na página 12.

SYKORA, H. T. et al. Stochasticdelaydiffeq. jl-an integrator interface for stochastic delay differential equations in julia. 2020. Citado na página [9](#).

TORVALDS, L. Linux: a portable operating system. Master' s thesis, University of Helsinki, 1997. Citado na página [6](#).