

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

Softwares Livres na Academia e na Indústria

Lorena, São Paulo
17 de junho de 2021

Resumo

Demonstrou-se como é possível construir uma série de aplicações baseada em softwares de licença livre, à partir de um sistema aberto, o Linux com interface EXWM - Emacs X Window Manager. Além disso, foi propiciado casos reais de aplicações na Indústria e no investimento privado, autônomo. Bem como, utilizações na Academia, à nível de lecionar, e pesquisa. Sustenta-se que a economia aberta possui similaridade estrutural ao movimento Open Source e seu desenvolvimento, o que aponta que essa é e continuará a ser, paulatinamente mais, o paradigma de desenvolvimento econômico tecnológico. Assim, imprescindível à formação do engenheiro.

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

Lista de ilustrações

Figura 1 – Genealogia Distribuições Linux	7
Figura 2 – EXWM - Emacs X Window Manager	8
Figura 3 – Diagrama esquemático simplificado - comunicação cliente-servidor	9
Figura 4 – Diagrama esquemático - comunicação cliente-servidor	9

Sumário

1	INTRODUÇÃO	5
1.1	Objetivo	5
2	REVISÃO BIBLIOGRÁFICA	6
2.1	Open Source	6
2.1.1	Diversidade	6
2.2	O Linux	7
2.2.1	Origem Histórica	7
2.2.2	O Emacs	8
2.3	Aplicações na Indústria	8
2.3.1	Freqtrade	8
2.3.2	OR-Tools	10
2.4	Aplicações Acadêmicas	10
2.4.1	DifferentialEquations.jl	10
2.4.1.1	Portabilidade em Julia	11
2.4.1.2	Portabilidade em Python	11
2.4.1.3	Portabilidade em R	11
2.4.2	O L ^A T _E X	12
2.4.2.1	Classe Canônica ABNT de produção científica	12
2.5	Trabalhos Canônicos na Área de Computação	13
2.5.1	Structure and Interpretation of Classical Mechanics (SCIM)	13
2.5.2	Structure and interpretation of computer programs (SICP)	13
2.5.3	SICMUtils - Portabilidade de (SCIM) em Clojure	14
3	MATERIAIS E MÉTODOS	15
3.1	Convite ao orientador	15
3.2	Delineação do tema	15
3.3	Organização cronológica	15
3.4	As fases da escrita da Monografia	16
3.5	Junção de notas	16
3.6	Pesquisa bibliográfica	16
3.7	Revisão contínua	16
3.7.1	Organização da apresentação	17
4	RESULTADO E DISCUSSÕES	18
4.1	Convite ao orientador	18

4.2	Delineação do tema	18
4.2.1	Da Indústria	18
4.2.2	Da academia	19
4.2.3	A intersecção	19
4.3	Organização cronológica	20
4.4	As fases da escrita da Monografia	20
4.5	Junção de notas	20
4.6	Pesquisa bibliográfica	20
4.7	Revisão contínua	21
4.8	Organização da apresentação	21
5	CONCLUSÃO	22
	REFERÊNCIAS	23

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 os 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 pétreas [seção 2.1](#), garantindo os direitos de estudo, cópia, modificação e redistribuição.

Bem como a ciência se beneficia com seus rápidos avanços, de uma comunidade global de participantes, com as mais distintas especializações profissionais. Também, beneficia-se a computação com a comunidade aberta, e especialização eclética, tanto de membros quanto de softwares.

1.1 Objetivo

Demonstramos a defasagem que um profissional de engenharia apresentaria, sem forte formação dentro da computação. Ademais, ao partir da gama de aplicações, em estado da arte, as quais são partilhadas de forma aberta e livre, pretende-se reinterar o caso da necessidade de compreensão do fenômeno dos softwares abertos. Pois, essa lógica e dinâmica não possui paralelos nem na economia, nem na comunidade científica ([HIPPEL; KROGH, 2003](#); [PETERS, 2009](#)).

Há até 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. Porém, mal compreendido e, raramente, debatido sob esse prisma, dentro das ciências sociais. Cita-se o mais citado dos artigos na busca no Google Scholar “The future(s) of open science”, o qual somente cita três vezes o termo Open Source, em tom dismissivo ([MIROWSKI, 2018](#)). Assim, argumenta-se que a formação básica do engenheiro na área da computação precisa ser sólida, bem como o entendimento das forças que moldaram esse movimento, de forma a poder entender o futuro em que caminhamos, de forma crítica.

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.

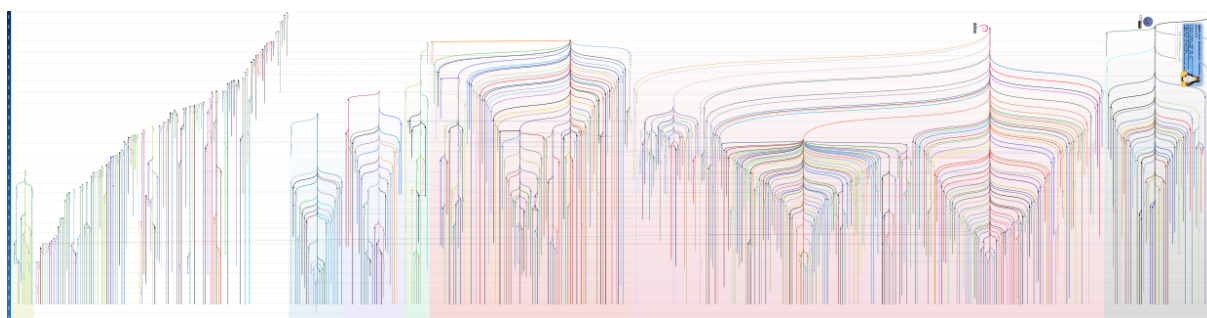
Assim, bem como em qualquer outro escopo de software, a variação dos softwares abertos e livres (FOSS) sempre serão superiores aos monopolizados.

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](#)).

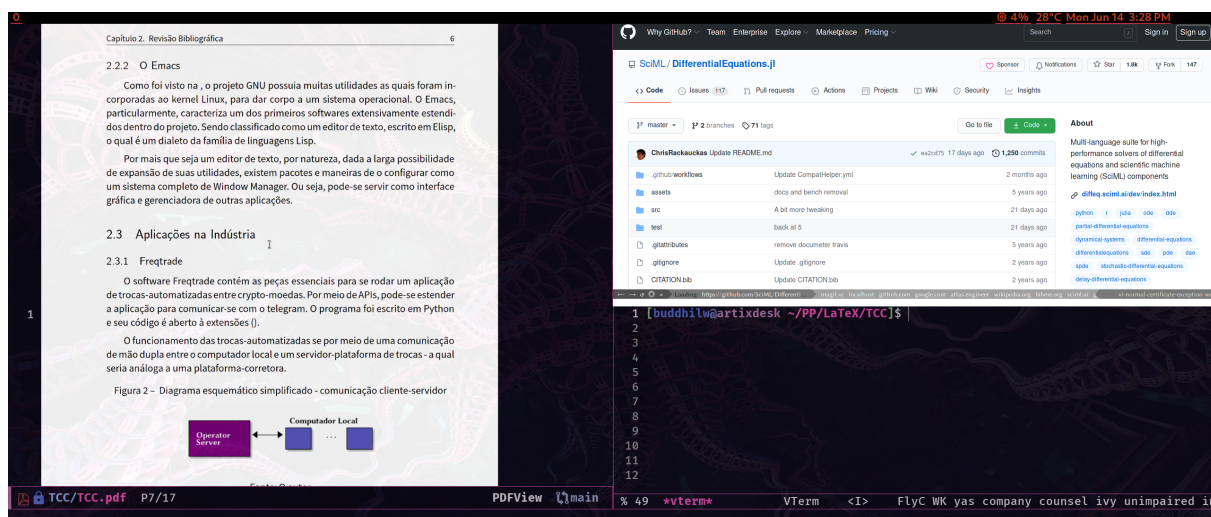
Paralelamente, Linus Torvalds desenvolveu um sistema operacional portátil aberto, como sua tese de mestrado ([TORVALDS, 1997](#)). Por fim, houve uma junção dos projetos, os quais colaboravam o Linux, como sistema operacional, e GNU com todas as aplicações utilitárias do sistema ([STALLMAN, 1997](#)).

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.

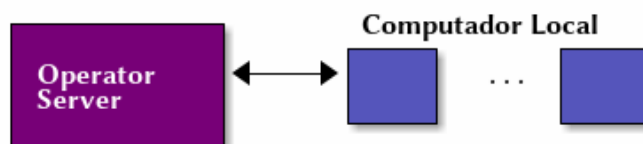
2.3 Aplicações na Indústria

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 contriuidores, 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.

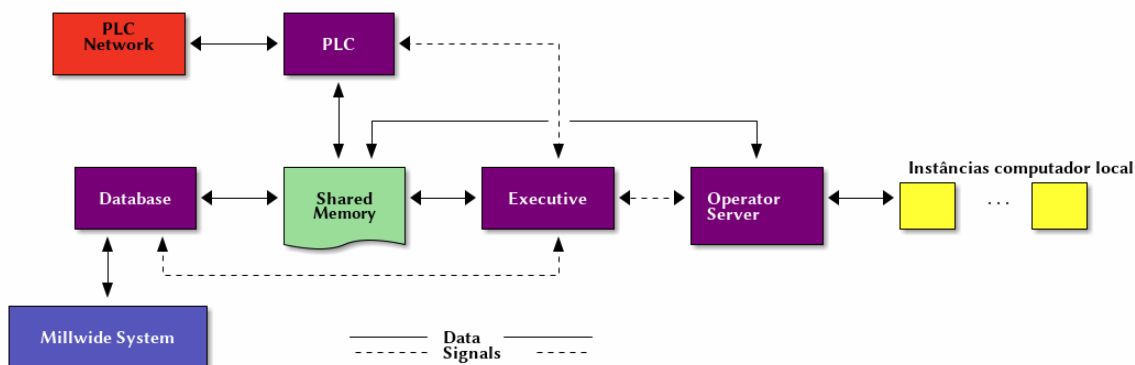
Figura 3 – Diagrama esquemático simplificado - comunicação cliente-servidor



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

O planejamento da monografia seguiu as seguintes etapas:

1. Contato e estratégias com o orientador;
2. Determinação do tópico principal;
3. Delineação específica dos subtópicos;
4. Organização da agenda à partir das datas finais objetivadas;
5. Início à escrita do TCC;
6. Junção de notas exploratórias sobre o tópico;
7. Pesquisa bibliografia;
8. Escrita de todos os tópicos da monografia;
9. Revisão contínua à cada etapa da monografia;
10. Organização de apresentação;

3.1 Convite ao orientador

Procurou-se um docente tanto familiar ao autor, quanto versado e interessado no tópico de pesquisa.

3.2 Delineação do tema

Dado que o tema softwares livres na indústria e academia é um tópico indefinidamente abrangente, houve uma necessidade imprescindível de delimitar gradua-mente o tópico.

Nas revisões contínuas, determinava-se o quão profundo tinha se abordado os tópicos, e se era necessário dar procedência, ou seguir o andamento de outras parte da agenda.

3.3 Organização cronológica

Decidiu-se por seguir o método de diagrama de Gantt, amplamente utilizado em organização de projetos. As partições temporais foram ponderadas pelas datas limites objetivadas.

3.4 As fases da escrita da Monografia

Como a monografia se caracteriza como uma Análise Exploratória, o enfoque foi na literatura concernente ao Tema.

Na subtópico acadêmico, procurou-se dar enfoque ambos softwares auxiliares de pesquisas, quanto úteis à sala de aula. Bem como, pretendia-se explicar conceitos-chave na compreensão da dinâmica histórica e atual do ecossistema em que se desenvolve e aplicam softwares.

Ademais, nas aplicações tecnológicas, o autor documentou softwares que já utilizou em situações reais.

3.5 Junção de notas

Ao se utilizar os softwares os autores havia produzido anotações sobre o desenvolvimento do software. Uma auto-documentação, enquanto se desenvolvia, ou se estudava os mecanismos de uma aplicação. Assim, utilizou-se dessas notas para construção da monografia.

3.6 Pesquisa bibliográfica

A revisão da bibliografia objetiva facilitar o entendimento da indústria dos softwares, a qual é difere em seu aspecto majoritariamente livre e aberto. Além do mais, procurou-se fazer um recorte de história, usos e desenvolvimento em estado da arte de diversos softwares icônicos.

A maior quantidade de informações históricas encontra-se armazenada digitalmente, em sites de instituições internacionais, como o GNU. Conquanto, nas aplicações procurou-se o embasamento em artigos científicos.

Utilizou-se do Google Scholar e a plataforma da ACM (Association for Computing Machinery), da qual o autor é membro e na qual existe material especializado em computação e softwares.

3.7 Revisão contínua

Parte fundamental da organização da monografia foi a troca de informações, auxílio, e integração constante das considerações do orientador Dr. Wei-Liang.

3.8 Organização da apresentação

Na apresentação, procura-se utilizar ferramentas interativas e demonstração de conceitos por meio de programação em tempo real aplicações. Escolheu-se utilizar da ferramenta Org-mode, integral ao Emacs.

4 Resultado e Discussões

4.1 Convite ao orientador

O autor já havia interagido com o Dr. Wei-Liang, o qual no passado havia pedido, a título de pesquisarem juntos, que se instalasse o sistema operacional do Linux. Ademais, uma das aplicações era na resolução e simulação de sistemas resolvidos por Lagrangeanas - tópico lecionado pelo orientador, na disciplina de Mecânica Clássica.

Assim, fortunamente o orientador se sentiu inclinado a participar desse trabalho. E, concorda que o ensino sobre a iniciativa Open Source, bem como os softwares em si, são detrimenais na vida acadêmica.

4.2 Delineação do tema

A iniciativa GNU e o open source é indefinidamente abrangente. Existem softwares abertos escritos para virtualmente qualquer outra área do exercício cognitivo do ser humano.

Necessitou contentar-se em parcialmente mostrar aplicações mais em mão, as quais seriam mais relevantes ao escopo da monografia. Bem como, optou-se por aprofundar-se em tópicos já utilizados na vida do autor para se resolver problemas na academia e na indústria.

Assim, a delimitação principal se deu entorno de ferramentas abertas das quais, na indústria foram apresentados OR-Tools e Freqtrade.

4.2.1 Da Indústria

As línguas utilizadas para o desenvolvimento desses softwares foram Python e C++. Porém, no caso de OR-Tools existe portabilidade para C++ (nativo), Python, Java, .NET. Todos os projetos e línguas possuem licença aberta.

OR-Tools destina-se a resolver um problema virtualmente conspícuo a qualquer negócio ou estrutura social prestadora de serviço. Pois, o problema de agendamento de horários, com condições de restrições impostas, é extremamente comum. Por exemplo, para organizar turnos de funcionários, em uma empresa; turnos de enfermeiros em um hospital - exemplo dado na própria documentação do software etc.

Freq-trade destina-se a automação de investimentos em crypto-moedas. A ferramenta surgiu espontaneamente da comunidade aberta, sem incentivo algum externo - até, pois, não existe um responsável que chefia o projeto. E, num curto período de desenvolvimento, cinco anos, desde 2017, a aplicação conta com inúmeras otimizações computacionais, e APIs com interfaces como Telegram. Não existe paralelo de software fechado o qual é comercializado, com mesmo escopo.

4.2.2 Da academia

Optou-se a estudar aplicações escritas em Python, Julia, Clojure(script). Pois, o autor possui histórico de utilização dessas ferramentas. Com efeito, a pesquisa constituiu, em grande parte, na junção de todos os esparços projetos em que o autor já havia colaborado ou desenvolvido. E, no aprofundamento da compreensão de todas esses projetos, para que fosse possível fazer uma apresentação coesa do ferramental utilizado.

Os programas em Julia possuíam cunho numérico. Pois, sua computação otimizada compara-se a performance de FORTRAN e C++. Assim, o principal pacote estudado foi DifferentialEquations.jl, a qual pode ser utilizada para resolver uma gama de equações diferenciais, incluindo a categoria Estocástica.

O programa em Python é de processamento de imagens. Escolheu-se essa aplicação, pois demonstra o quão abrangente é a portabilidade de programas escritos em outras línguas ao Python - principalmente, devido a grande comunidade de cientistas e programadores que o utilizam.

4.2.3 A intersecção

Com Clojure(Script), Bash e \LaTeX , foi possível criar tanto simulações que estendem a compreensão de fenômenos físicos do campo abstrato ao visual. Também, empregou-se de Closh e Babashka, duas bibliotecas de Clojure dedicadas a interoperabilidade com Bash, a qual se processava milhares de arquivos, por meio de pipelines de arquitetura complexas. Assim, facilitando a escrita de uma quantidade enorme de relatórios técnicos na área de segurança do trabalho.

Quanto a Clojure(Script), foram pesquisadas simulações gráficas, as quais inerentemente são simples de se programar em Clojure. Pois, seu ecossistema pode interoperar entre JVM (Java Virtual Machine) e JavaScript. Assim, dado que ambos Java e JavaScript foram as línguas mais populares por décadas, existem uma enorme quantidade de ambientes e bibliotecas a nossa disposição. Bem como, Clojure é uma Lisp - e, caracteristicamente, essa família é utilizada para prototipagem, vide AutoCAD escrito em AutoLisp.

Dado que existem bibliotecas dedicadas a Mecânica Clássica em nível de mestrado, bem como existe capacidades de se renderizar conceitos físicos abstratos. Então, decorreu-se a computação de programas que demonstravam conceitos de Física, Cálculo, e Computação Generativa.

4.3 Organização cronológica

A organização do projeto se deu utilizando os diagramas de Gantt, e as agendas dinâmicas, `/org-agenda/`, nativas do Emacs. Por conseguinte, toda a comunicação sobre o projeto foi materializado por meio de cronogramas e agendas.

4.4 As fases da escrita da Monografia

Iniciou-se a escrita da monografia pela pesquisa bibliográfica, dado que a pesquisa se categoriza, epistemologicamente, como uma Análise Exploratória. O template utilizado para escrita do projeto foi o `abnTEX2`, biblioteca do `LATEX` dedicado a escrita de documentos sob as normas ABNT.

4.5 Junção de notas

O `org-mode` é uma ferramenta de organização e anotação, desenvolvida para pesquisa e reprodutibilidade de pesquisas. Assim, todas as etapas de desenvolvimento do projeto foi desenvolvido e documento utilizando essa ferramenta.

A documentação implícita ao desenvolvimento é uma característica fundamental e pensada do `Org-mode`. Pois, segue o paradigma de `Literate Programming`. Assim, há uma mistura de notas e códigos em um mesmo ambiente. Fazendo que a prototipação do código seja automaticamente documentável.

Por fim, é possível exportar qualquer anotação de extensão `.org` para `.tex` e consequentemente `.pdf`. Assim, reutilizar dessas anotações foi uma tarefa natural.

4.6 Pesquisa bibliográfica

A pesquisa bibliográfica foi em grande parte encontrada em sites sem autores. Pois, a documentação foi criada ou pela comunidade, ou por instituições, não unicamente publicada em formato de livro ou artigo.

Da parte acadêmica, como o pacote de resolução de equações diferenciais, em Júlia, encontrou-se vasta literatura. Até porque, a língua foi criada para resolver problemas computacionais tocantes principalmente a comunidade científica do MIT.

Por fim, algumas ferramentas apenas existem documentadas e contidas inteiramente em repositórios, como o Github. Foram os casos do OR-Tools e as simulações em Clojure. São publicações puramente eletrônicas.

4.7 Revisão contínua

Dentro do cronograma concebido para desenvolvimento da monografia, pôs-se metas e etapas de escrita do documento. E, estipulou-se os conteúdos necessários para cada nova versão do documento, aliado a uma data limite. Assim, o retorno do orientador ao orientado pode se dar mais frequentemente.

4.8 Organização da apresentação

Como se está utilizando de diversas ferramentas e extensões dentro do Emacs e do Org-mode, em particular, optou-se por utilizar uma de suas funcionalidades dedicada a apresentações.

5 Conclusão

Dentro das iniciativas Open Sourced, existem virtualmente aplicações para todas as áreas das aventuras do ser humano. Isto é, seja o uso de computação gráfica, processamento de imagens, expressões artísticas, ou ferramentas de organização, há um software escrito para esse fim.

É observável a falta de competição quanto a quantidade e qualidade de softwares fechados e abertos. Ademais, a capacidade de extensão dos softwares livres, por meio das quatro fundamentais liberdades de suas licenças, permitem que exista uma variedade imensa de uso de uma mesma ferramenta. Por fim, o enriquecimento dessa participação colaborativa solidifica e generaliza as ferramentas abertas - num grau e rapidez inimagináveis, comparados ao processo de desenvolvimento por uma equipe fechada.

Demonstrou-se diversos usos desses software, sem paralelos, na Indústria e na Academia. Ambos possuem algoritmos na fronteira da ciência, a qual o uso apenas é limitado pela imaginação e capacidade formativa do usuário-desenvolvedor. Assim, é imprescindível que no curso de engenharia se tenha uma ampla formação com base nos softwares livres, sob os quais a computação moderna subiste.

Apontou-se algumas portas de entrada, fundamentais, a esse paradigma, os sistemas com Kernel Linux, as aplicações GNU as quais lhe dão roupagem, e por fim os sistemas sociais onde se desenvolve os softwares, como a plataforma de versionamento Git/Github. Ademais, demonstrou-se, com efetividade, que é possível construir e rodar todas essas aplicações num sistema totalmente aberto.

Referências

- ABELSON, H.; SUSSMAN, G. J. Structure and interpretation of computer programs. [S.l.]: The MIT Press, 1996. Citado na página 13.
- 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 12.
- ARAUJO, L. C. A classe abntex2. Documentos técnicos e científicos brasileiros, 2018. Citado na página 12.
- EMACS History. 2021. <<https://www.emacswiki.org/emacs/EmacsHistory>>. Citado na página 7.
- FANG, F. et al. Cryptocurrency trading: a comprehensive survey. arXiv preprint arXiv:2003.11352, 2020. Citado na página 8.
- GOWDA, S. et al. Sparsity programming: Automated sparsity-aware optimizations in differentiable programming. 2019. Citado na página 10.
- 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 14.
- HARALAMBOUS, Y. Fonts & encodings. [S.l.]: ” O’ Reilly Media, Inc.” , 2007. Citado na página 12.
- 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 5.
- KNUTH, D. E. The art of computer programming, vol 1: Fundamental. algorithms, p. 187, 1968. Citado na página 5.
- KNUTH, D. E. TEX: the Program. [S.l.]: Addison-Wesley, 1986. Citado na página 12.
- LAMPORT, L. LATEX: a document preparation system: user’ s guide and reference manual. [S.l.]: Addison-wesley, 1994. Citado na página 12.
- LINUX, list of distributions. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/List_of_Linux_distributions>. Citado na página 7.
- MA, Y. et al. ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling. 2021. Citado na página 10.
- MACOS version history. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/MacOS_version_history>. Citado na página 6.

- MICROSOFT, list of operating systems. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/List_of_Microsoft_operating_systems>. Citado na página 6.
- 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 5.
- 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 5.
- RACKAUCKAS, C. et al. Diffeqflux.jl-a julia library for neural differential equations. arXiv preprint arXiv:1902.02376, 2019. Citado na página 10.
- 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 10.
- RACKAUCKAS, C. et al. Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385, 2020. Citado na página 10.
- 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 10.
- 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 10.
- 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 10.
- 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 10.
- SMITH, C.; RITCHIE, S. SICMUtils: Functional Computer Algebra in Clojure. 2016. <<http://github.com/sicmutils/sicmutils>>. Citado na página 14.
- STALLMAN, R. Linux and GNU - GNU Project - Free Software Foundation. 1997. <<https://www.gnu.org/gnu/linux-and-gnu.html>>. Citado na página 7.
- 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 7.
- SUSSMAN, G. J.; WISDOM, J. Structure and interpretation of classical mechanics. [S.l.]: The MIT Press, 2015. Citado na página 13.

SYKORA, H. T. et al. Stochasticdelaydiffeq. jl-an integrator interface for stochastic delay differential equations in julia. 2020. Citado na página [10](#).

TORVALDS, L. Linux: a portable operating system. Master' s thesis, University of Helsinki, 1997. Citado na página [7](#).