PEDRO GOMES BRANQUINHO

Free software in industry and academia

Lorena

2021

PEDRO GOMES BRANQUINHO

Free software in industry and academia

This monograph is presented to the Engineer School of Lorena, the University of São Paulo, to obtain the title of Bachelor by the Graduation Program on Engineering Physics with emphasis on the Science of Materials.

University of São Paulo - USP
Engineer School of Lorena

Monograph, Final Paper

Supervisor: Dr. Wei-Liang Qian (钱卫良)

Lorena
2021

PEDRO GOMES BRANQUINHO
Free software in industry and academia/ PEDRO GOMES BRANQUINHO. – Lorena, 2021-
Supervisor: Dr. Wei-Liang Qian (钱卫良)

Dissertation – University of São Paulo - USP
Engineer School of Lorena
Monograph, Final Paper , 2021.

1. Free Software. 2. Open Source. 2. Academy and Industry. I. Wei-Liang Qian. II. University EEL-USP. III. Escola de Engenharia de Lorena. IV. Free Software on Industry and Academia.

To those whom found me in their own path
and, by finding me, made part of my own.

Acknowledgements

My acknowledgments wouldn't fit on a single page. But, for conciseness, I will mention those who are closer to my work. I thank first my advisor, Wei-Liang Qian, who, in his patience and kindness, knew how to conduce me to produce the present work. I thank Juan Zapata for the support and enthusiasm in teaching Mathematics and showing me the way of self-study. Also, Antônio Jefferson Machado who has ministered Solid State Physics classes in which I learned a lot about being an engineer, the roles of the scientific community in furthering the high-end applications related to matter. Last but not least, I thank Luiz Eleno, who has been a role model, and has taught me so much about computation throughout the years.

And, in a big umbrella, I thank all those anonymous people who have contributed to my experience of communal sharing and understanding in the Open Source community. Especially, David Wilson, the founder of Systems Crafter, from whom I derived the basis of my Emacs's system. Thank you, all the members, peers, professors, and staff of EEL-USP, in which I spent a great part of my current life.

"There is no one unique truth concerning the properties of sets: Rather, the reality of what a set is bifurcates into several alternative realities, all equally plausible and all equally true.

"Lastly, it has been shown that we will never have absolute certainty that set theory—or mathematics generally—is free of contradictions. It is not merely a question of the state of current knowledge: Rather, what has been shown is that it is fundamentally impossible ever to prove the consistency of mathematics.

"(...) It is already a remarkable fact that animals (including homo sapiens) are able to abstract out of experience the fact that there is such a thing as a cluster, a batch, a bundle of similar objects—and that such a bundle is a separate ***unit of reality****. (...) That kind of iterated abstracting seems to be the essence of the human intellectual enterprise. More than that, it is perhaps the long-term ecological function of brains.*

(Charles C. Pinter, A Book of Set Theory, p.311)

Abstract

In this monograph, we discuss key concepts when working with modern Free and Open Source Software (FOSS) applications. These applications were chosen in contexts of both Academy and Industry. The development-environment to run these applications was a FOSS-based system, which primarily consists of Emacs X's Window Manager (EXWM) and Artix Linux OS. An explanation for motives, and how often FOSS is currently adopted, is given when we deal with the basic notions and the ongoing trend for adopting FOSS. Some canonical works on Scientific Computing - SCIM and SICP - are considered and their relevance investigated. We use these examples, as motives for giving emphasis to the role of simulations and how they can enhance physics-knowledge acquisition. Regarding application development, the discussions are carried out by focusing on some specific tools, which should be taken as formal examples. Their potentiality, as particular tools, for particular problems, can be generalized - given that we chose any other field of inquiry and find or extend suited applications built for this field. In particular, we elaborate on these applied FOSS: Freqtrade, OR-Tools, DifferentialEquations.jl and Clojure2D. Showing, thus, that once one have a suitable environment to run these types of application, then the easiness to use and extend them follow.

Key-words: modern application. academy. industry. foss. physics-knowledge acquisition. scientific computing. simulations. freqtrade. or-tools. differentialequations.jl. clojure2d.

Resumo

Nessa monografia, discutimos conceitos teóricos chave, os quais devem ser considerados quando se trabalha com aplicações modernas. As aplicações discutidas no documento contextualizam-se tanto a Indústria, quando a Academia. Essas aplicações foram rodadas sob um sistema de desenvolvimento baseado em softwares livres e abertos (FOSS). O ambiente consiste primariamente de EXWM, Aritx Linux OS. As noções básicas e a presente tendência de adoção de FOSS em contexto acadêmico e em práticas industriais são brevemente revistas. De forma a explicar alguns dos motivos e quão frequentemente FOSS são atualmente empregados. Alguns trabalhos canônicos de Computação Científica - SCIM e SICP - são considerados, e suas relevâncias investifadas. Isto serve de motivação para enfáse em utilização de simulações como meio de aprendizado de física. No tocante ao uso de aplicações, a discussão é feita de forma a focar em apenas em algumas ferramentas específicas, as quais devem ser tomadas como exemplos formais. As potencialidades como ferramentas particulares podem ser generalizadas. Basta-se escolher qualquer outro campo de pesquisa e utilizar-se de aplicações escritas a ele. Em particular, focamos em trabalhar com os seguintes FOSS: Freqtrade, OR-Tools, DifferentialEquations.jl e Clojure2D. Mostrando, assim, que uma vez que temos um sistema ajustado a rodar esse tipo de aplicação, por conseguinte segue uma facilitação em utilizar e estender essas aplicações.

Palavras-chaves: aplicações modernas. contexto acadêmico. contexto industrial. foss. aprendizado de física. computação científica. simulações. freqtrade. or-tools. differentialequations.jl. clojure2d.

List of Figures

List of abbreviations and acronyms

FOSS	Free and Open Source Software
OSS	Open Source Software
IDE	Integrated Development System
abnTeX	ABsurdas Normas para TeX
OR-Tools	Operational Reseach Tools
IT	Information Technology
GUI	Graphical User Interface
LCS	Laboratory of Computer Science
GNU	GNU's Not Unix
WM	Window Manager
EXWM	Emacs's X Window Manager
SCIM	Structure and Interpretation of Classical Mechanics
SICP	Structure and interpretation of computer programs
OS	Operational System
PDF	Portable Document Format

Contents

1 Introduction

Physics Engineer, by definition, is a generalist professional. Among other valuable experiences that can be acquired through the training, the use of free and open-source software (FOSS) and participation in the open community constitute two essential pieces.

The variability, which open source software (OSS) brings to existing applications and their extension, can change altogether user's experience. Thus, taking him closer to acting as a developer. This experience of interloping user and developer roles doesn't require that you are a computer scientist or a Information Technology (IT) professional by training. For, programming can be seen as both a Science and an Art (KNUTH, 1968) - e.g., an exercise of self-expression.

OSS guarantees four fundamental liberties section 2.1, the right to study, copy, modify and redistribute it.

Just as the scientific enterprise benefits, with it's rapid development, by means of the global community's participation, which holds space for individuals with a variety of different training. Also, the computation enterprise benefits from the variety of people's training, which constitute the body of the open source community.

1.0.1 Considerations on Mathematics and Physics historical developments

As has been pointed out by Bourbaki, when dealing with theories utilizing mathematical formalization,

> (...) The only important point is the correct observance of the rules of syntax. Thus, as everyone knows, the same algebraic calculation can be used to solve problems about pounds weight or pounds sterling, about parabolas or motion under gravity. The same advantage attaches to every text written according to the axiomatic method, and for the same reasons: once the theorems of general topology have been established, they may be applied at will to ordinary space, Hilbert space, and many others (BOURBAKI, 2004).

Also, in the same text, Bourbaki emphasizes that intuition can greatly benefit from comprehending these symbolical results.

> This faculty of being able to give different meanings to the words or prime concepts of a theory is indeed an important source of enrichment of the mathematician's intuition, which is not necessarily spatial or sensory, as is sometimes believed, but is far more a certain feeling for the behaviour of mathematical objects, aided often by images from very varied sources, but founded above all on everyday experience (BOURBAKI, 2004).

Just as mathematicians would benefit from generalizing their daily experience, through formalization. Also, could be argued, the physicist and then the engineer physicist learning physics benefit from formalization. Because, modern physics is essentially tied to mathematical formalization.

> "In this book (*Mécanique analytique*), he lays down the law of virtual work, and from that one fundamental principle, by the aid of the calculus of variations, deduces the whole of mechanics, both of solids and fluids.
>
> (...) Lagrange remarked that mechanics was really a branch of pure mathematics.
>
> (...) It is said that he prided himself that from the beginning to the end of the work there was not a single diagram." (PANZA, 2003)

Concatenate the fact that the developer of Classical Mechanics saw it as an branch of mathematics. And, then that current notation of Quantum Mechanics is based on the work of Paul Dirac on Lagrangians(DIRAC, 2005)

> (...) the whole of the classical Hamiltonian theory, which is just a structure built up on this notion, could be taken over in all its details into quantum mechanics.
>
> (...) We must therefore seek our quantum Lagrangian theory in an indirect way. We must try to take over the ideas of the classical Lagrangian theory, not the equations of the classical Lagrangian theory. (DIRAC, 2005)

That is, the *logical structure* on which the classical Lagrangian theory was formulated upon. i.e., It's *meta-mathematical* arguments.

When Cantor was working on highly technical papers on trigonometrical series, he realized his operations on series could be formalized in term of sets. And, then started the revolution on the foundations of mathematics (PINTER, 2014).

> In the course of the past seventy years or so, set theory has come to be widely recognized as the fundamental, "unifying" branch of mathematics. We have already seen how the natural numbers can be constructed, and their properties derived, within the framework of set theory; from there, it is easy to develop the rational numbers, the real and complex numbers, as well as remarkable systems such as Cantor's "transfinite cardinals." The notions of function, relation, operation, and so forth are easily defined in terms of sets, and, as a result, every known branch of mathematics can be formulated within set theory. (PINTER, 2014)

So, just as mathematicians benefited from using axiomatic development to formalize mathematics. And, just as physicists benefited on using the mathematical formalization to develop their model of reality. All these development, although anchored in empirical necessities, one lesson can be learned, in general,

> Thus one is often led to study with profit those parts of a theory which traditionally have been neglected in this theory but studied systematically in a general axiomatic context, of which the given theory is a special case (BOURBAKI, 2004).

1.1 Objectives

The authors propose a development from general theory to special case; from the modeling theory of computation to modern applications. Until the end of the work we wish to show the thesis that "Modern applications can be used to elicit physics knowledge from simulation. Furthermore, these applications have a good fit under FOSS based systems."

Thus, the scheme of the current work is divided as follows,

1. Exposition to key concepts of a FOSS environment [Bibliographical review].
2. GNU/Linux.
3. Compilers.
4. Towers of interpreters.
5. Why and how do someone inserts himself into a FOSS-based environment [Bibliographical review].
    a) Bibliographical evidence of how FOSS adoption in Industry and Academia.
    b) What benefits may follow from FOSS adoption.
6. Show the functionality of applications concerning Academia and Industry. [Literature review/Results and Discussion].
    a) Academic [Literature review/Results and Discussion]
        i. Example of well established programs used in teaching graduate level physics (SCIM and SCIP) [literature review].
        ii. Extension of different graphical and numerical libraries available as FOSS to understand math and physics [Results and Discussion].
    b) Industry [Results and Discussion]
        i. OR-Tools
        ii. Freqtrade
    c) Both Industry and Academic - Organizational tools [Material and Methods/Results and Discussion]
        i. Org-mode
        ii. Org-agenda
        iii. LaTeX charts
        iv. Typography of the present work.

Lastly, we conclude by synthesizing what has been understood from the current work.

1.2 The interconnection between Applications and the OS

The author used an **Application** as to comprise any end product of software development.

1.2.1 Why does GNU/Linux matter?

We will discuss, as a brief introduction, what is the Operational System (OS) GNU/Linux, and why it's the *de facto* opening door to the Open Source Community. Firstly, the GNU/Linux is the first and most successful project carried out in the Open Source paradigm (TU et al., 2000; WEST; DEDRICK, 2001). Therefore, it's use is a way to acquaintance, in practical terms, with how a business dependent on mainly using the open sourced development products might operate (FINK, 2003). Furthermore, the new user-developer of the GNU/Linux framework must inherently learn about the accompanying software which comes with it's distribution - which increases it's chance of adoption (WEST; DEDRICK, 2001). This way, the user gradually becomes accostumed to participate on development and extend programs (HERTEL; NIEDNER; HERRMANN, 2003).

Beyond it's initial philosophical appeal to user liberty, GNU/Linux is today's standard in Technological Enterprises. And, although fundamentally opposed of many age-old *modus operandus* of the *status quo* of companies, there is no doubt left under analysis of the benefits it brings to the companies, general economy, and the society triad (MOODY, 2009). To make use of GNU/Linux, thus, is a way to be inserted in this new economic paradigm (HIPPEL; KROGH, 2003; PETERS, 2009).

1.2.2 How high level applications benefit from an OS

In the hierarchy of software and applications, the Operational Systems (OSes) can be seen as a meta-application or meta-software.

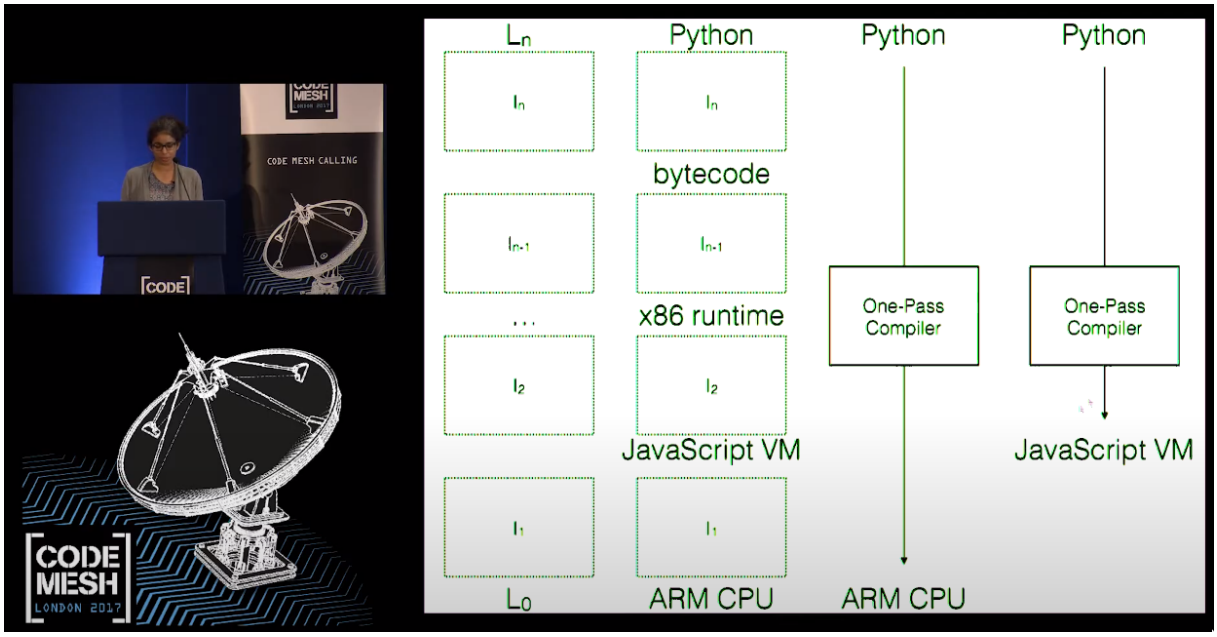
"The evaluator, which determines the meaning of expressions in a programming language, is just another program." (ABELSON; SUSSMAN, 1996)

There exists levels, or layers, of abstractions in virtually any application. That is, the concept of meta-programming and Towers of Interpreters comprise a common situation, for which a devoted field of study exists. Thus this area has direct implication for software development practice, as it's a ubiquitous problem faced in computing.

Any OS, as the GNU/Linux, comprise an essential layer in this tower of interpreters. Particularly, an OS communicates with *firmwares* - low-expressivity and highly-performing software, which control *hardware*. Also, they communicate with high-expressivity software, among which contain the user-developer written or extended software. There-

fore, the OS play a fundamental role, mediating between low and high level software. This function categorizes them as a *middleware* software.

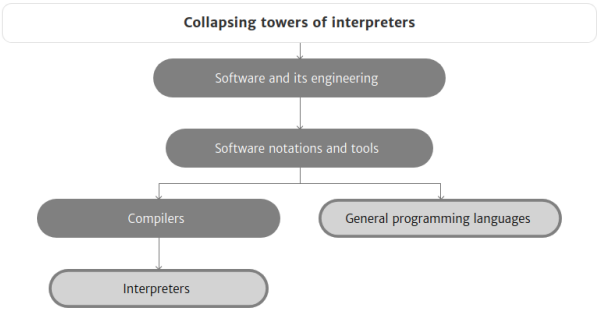
Figure 1 – Schema of a tower of interpreters



Code Mesh, presentation "Towers of Interpreters", by Nada Amin

The characteristic problem of concatenate a system of software, one on top of the other, introduces complexity to maintaining compatibility among program's versions and it's performance. The study of these behavior and it's theoretical solutions posses a field of it's own. And, this field is autonomous, detached, for an example, from which languages compose the Tower of Interpreters; or which type of application we are dealing with (AMIN; ROMPF, 2017). The object of study is the final behavior of the system, and if it's a collapsible system.

Figure 2 – Categorization of the study of towers of interpreters

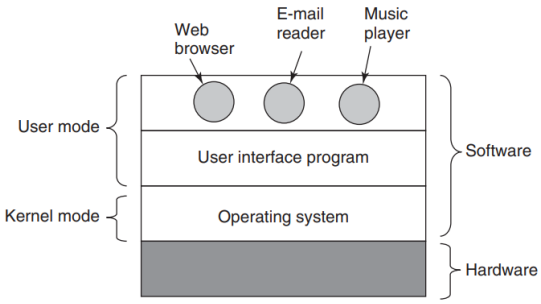


Reference: (AMIN; ROMPF, 2017)

Finally, the OSes consist in a big tower-collapser of interpreters. They are subordinate to collapsing firmware, middleware and high-level software. Therefore, as well as the OS

conduct this task, as much the user-developer experience is facilitated.

Figure 3 – Towers of interpreters and the Operational Systems.



Reference: (TANENBAUM; BOS, 2015)

> If every application programmer had to understand how all these things work in detail, no code would ever get written. Furthermore, managing all these components and using them optimally is an exceedingly challenging job. For this reason, computers are equipped with a layer of software called the operating system, whose job is to provide user programs with a better, simpler, cleaner, model of the computer and to handle managing all the resources just mentioned. (TANENBAUM; BOS, 2015)

1.3 On the influence of education in adoption

In the bibliographical literature, it's clear that the rate of OSS adoption in the Industrial sector - commonly refereed as"Production" - depends heavily on both: competency and the level of expertise a project require (LI; TAN; YANG, 2013; GALLEGO et al., 2015; SPINELLIS; GIANNIKAS, 2012).

At the same time, the adoption depends directly on the intrinsic inclination of the Informational Technology (IT) team (RACERO; BUENO; GALLEGO, 2021).

Therefore, regardless of how much a professionalizing course may increase the *intensity* and *quickness* of adoption. Data shows that, generally, those students and professionals positively correlated to *seeking autonomy* would adopt it (RACERO; BUENO; GALLEGO, 2020). This means that intrinsic motivation is key (GALLEGO et al., 2015). Nonetheless, it's important to notice that there exist a net-effect in adoption (SPINELLIS; GIANNIKAS, 2012). e.g., how much more peers adopt it, the more likely to any given individual to adopt it.

1.4 Performance and the current trend as reasons for adoption

Research with different areas of benchmarks state a performance gain, when utilizing GNU/Linux, compared to Windows (SULAIMAN; RAFFI, 2021). Although, even more

important, the key benefits are not in the differential performance *per se*, but in the training one naturally goes through upon utilizing a totally community-dependent Operational System.

Thereof, using GNU/Linux is a door for an individual professional shift. At the same time, this personal use increases the probability of adoption in whichever Industry carrer one may lead (HAUGE; SØRENSEN; CONRADI, 2008). Combined with that fact, the Industry per se has no observable effect on Open Source development of projects - as so far as measured in 2008 (HAUGE; SØRENSEN; CONRADI, 2008).

1.5 The set-conjunction between physics engineer and FOSS's users

We note that the deepness of training proposed, in graduation level, for a physics engineer makes them perfect candidates for the use of FOSS. Because, both trainings imply a *a priori* necessity for autonomy and purposeness (SCHRAPE, 2019; RACERO; BUENO; GALLEGO, 2020); imply a profound and will for generalistic technical knowledge (LI; TAN; YANG, 2013; GALLEGO et al., 2015).

1.6 How to leverage the potencial of OSS in Industry and Academia

In the present work, we utilized of many concept demonstrations, in which the autor has developed or/and extended applications, in a context of free and open source. Also, we present how can one collaborate in the community and how does that collaboration can imply significant professional connections. This way, both the so called "soft skills" and "hard skills" benefits have been elucidated, in practice.

2 Bibliography review

2.1 Open Source

Any program which permits the user-developer to have the following liberties:

1. The right to run the program, as seen fit, for any end.
2. The right to access the source code and study it.
3. The right to copy and redistribute it.
4. The right to modify the software.

Practically, the Open Source community fundamentally base itself upon the free distribution of it's tools and programs. One of the differential advantage of having innumerable other people extending the same software is that the advancement of the frontier of the program, in many directions, increases rapidly in relation to a program controlled by a limited number of programmers.

2.1.1 Diversity

Given that one fundamental right of OSS is the modification and propagation of new modified versions. This right implies in the observable wide range of maintained versions of these software, which doesn't have a parallel in any other technological enterprise.

For an example, one key application in any user's computer is a general Graphical User Interface (GUI)'s manager, commonly known as Window Manager (WM). These can be both Floating or Tilling, or mixed WM, e.g., Floating WM are those that the user must hover windows and adjust them manually; Tilling WM are those that a pre-defined program have a set of rules to resize automatically the windows in a screen.

While private Operational Systems (OS), as Windows and MacOS, have frequent releases - a total of twenty five releases for Windows. Generally, they've few *active* versions; Windows have currently four (MICROSOFT..., 2021). MacOS also have four active versions (MACOS..., 2021).

The fact there are only narrowly supported versions is due to, among many contributing factors, users lack the right to alter and extend the software's behavior. Therefore, they fall victims of discontinued support and restrictive access to the company's official upgrades.

On the other hand, there exists, in parallel, around two hundred seventy eight available distributions of Linux (LINUX..., 2021). Of which, there are main/root distributions, which each embody a set of different principles; theoretical and practical philosophies of

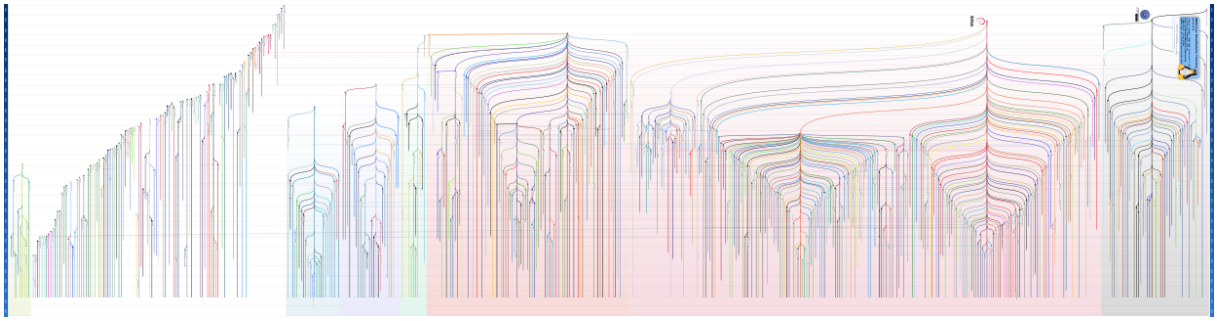
how to extend software.

Thus, just as with any other scope of software, the variability of FOSS always will be grater than monopolized ones.

2.2 GNU/Linux

There are root distributions of Linux, from which many other distributions emanate. Generically, these partitions are called families. We cite some of the most influential and popular ones, Red Hat Linux (), Debian (), CentOS (), Fedora(), Pacman-based (/), OpenSUSE (), Gentoo-based (), Ubuntu-based(), Slackware (), Open Sourced-based and the Independent Distributions (/).

Figure 4 – Genealogy of Linux's Distributions



Genealogical history of Linux Distributions (LINUX..., 2021)

2.2.1 Historical Origin

The GNU/Linux began as two separeted and different directions. GNU stands for "GNU's Not Unix", a recursive name. And GNU initialy has been developed as a collaboration of revolted academics by the restrictive secure system of the MIT Lab (Laboratory of Computer Science - LCS) (STALLMAN, 2002; EMACS..., 2021). Amongst them, there was the still active Richard Stallman, which heavily worked on the text editor of the time - already ten years into development. This editor became Emacs (EMACS..., 2021).

Parallel to these events, Linus Torvalds had been developing an open portatible operational system, as his master's thesis (TORVALDS, 1997).

Finally, both projects united in a symbiotic system, of which the OS was Linux (the formentined portible kernel) and the GNU's interface program whit all utilities one may have had in their computers at the time (STALLMAN, 1997)

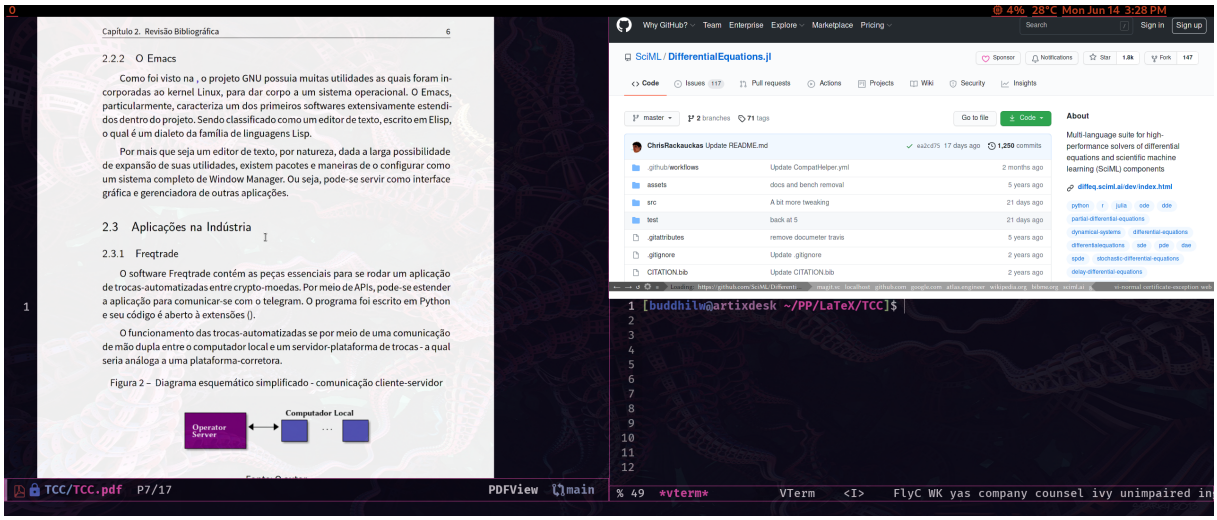
2.2.2 Emacs

As has been seen in , the GNU project already had developed a variety of applications, all of which, incorporated into Linux. This way, the OS gained a "body". The Emacs, particularly, characterize one of the first software extensively extended, as a project, in a open community.

Although, the label given to Emacs as an "editor" covers it's main function. Actually, the fundamental role of Emacs equates to evaluating Elisp expressions. Elisp as in a dialect of Lisp. Therefore, as an interpreter of a language, it has Turing complete capabilities. So it has a complete-system capability [1].

Even though Emacs usual use has been of an Integrated Development System (IDE), by it's unlimited potentiality and expressiveness, there exists packages and applications written in Elisp to become a fully featured Window Manager (WM). That is, Emacs, through Emacs's X Window Manager (EXWM), can serve as a graphical and manager interface to other applications.

Figure 5 – EXWM - Emacs X Window Manager



Source: author's WM ambient.

Figure 5 exemplifies a desktop environment which can render images, PDFs, Browsers et el, through Emacs.

[1] The GNU/Guix implementation of Linux, has been implemented in Scheme - a cousin with better performance than Elisp

2.3 Performance comparison among OS adoption

2.3.1 Performance

In user-level applications, the efficiency and processing benchmarks favor Linux over Windows (SULAIMAN; RAFFI, 2021).

In the same hardware, the underground/underlying programs maintaining a stationary Windows consume around 5% of CPU and 41% of RAM. In the mean time, Linux Mint - a popular distribution of Linux - consumed 1.8% of CPU and 24% of RAM. A pronounced difference of more than 200% in CPU use efficiency and approximately 200% in RAM efficienty (SULAIMAN; RAFFI, 2021).

A VBS-program script execution [2] amounts to a difference in time-execution of 4.249 seconds. In which, Linux Mint took 0.501 seconds, against a 4.75 seconds for Windows. E.g., there exist a roughly $\frac{4,75-0,501}{0,501} = 423\%$ in time-execution performance (SULAIMAN; RAFFI, 2021).

Lastly, there also exists other researches done with technical rigor in a broad spectrum of other applications and hardware. To name a few, in wireless (S.DEVAN, 2013); parallelism and management of server applications (AVELEDA et al., 2010); scientific programming (Fast Fourier Transform) et al (D'ELIA; PACIELLO, 2011); performance on Virtual Reality (VR) application (THUBAASINI; RUSNIDA; ROHANI, 2010) etc. All of which present different kinds of advantages for Linux-use compared to Windows. Although, against what is expected in view of Industry sector adoption, Linux has almost no difference, and sometimes can be worst depending of memory usage of the application (RISTOV; GUSEV, 2013), than Windows performance on multi-threaded and parallel systems, typically founded in cloud/cluster server use-case (AVELEDA et al., 2010).

2.3.2 FOSS adoption demography

Fritzgerald enunciated, in 2006, that the adoption profile of users of Open Source Software (OSS) passed through a phase "OSS 2.0". This phase was characterized as a transcendence from the mentality that the only people who would be able to use the system would be "hackers" (FITZGERALD, 2006). And, furthermore that such OS were even becoming "*mainstream*". That is, it became a common place in society and industry. He also professed the hypothesis that OSS project's activities were significant influenced due to big companies.

In 2008, another research tried to quantitatively verify the hypothesis. The data was obtained by questioners and interviews of Finland software industries. In contrast to the Fritzgerald's characterization, 50% of the companies utilized vastly FOSS, although, they

[2] An application concerning the labeled "Office" bundle.

had negligent impact of the development carried by the community. Lastly, more than 30% of these companies told that at least 40% of their companies value on software was due to services done by the community (HAUGE; SØRENSEN; CONRADI, 2008). Also relevant, these were mostly small to medium size enterprises.

In 2012, a study on big companies use of OSS - a thousand companies listed on US Fortune magazine - have derived some conclusions (SPINELLIS; GIANNIKAS, 2012):

- Adoption is directly related to IT's team need for expertise on the work they would partaken, as well as efficient cautiousness (GALLEGO et al., 2015; LI; TAN; YANG, 2013).
- Exponential growth on adoption, once the existence of an accumulation of users in the company.
- There exist a network effect and the effect is notorious in big companies.

Therefore, adoption in small, medium and big companies process vary, and there are no salient effects caused by companies on the expansion of open source projects. Nonetheless, these companies still benefit from the later (SPINELLIS; GIANNIKAS, 2012; HAUGE; SØRENSEN; CONRADI, 2008; FITZGERALD, 2006). Still, the most relevant positive-effect on the OSS and community is due to small enterprises (KSHETRI, 2004).

Regarding the expansion of the FOSS development paradigm and ecosystem, undoubtedly there is an grow and ubiquity of adoption (SCHMIDT, 2016). Moreover, FOSS has been considered in economical papers as the innovatory cradle of software industry (SCHRAPE, 2019; SCHMIDT, 2016).

More recent studies, dating to 2021 and 2020, determine that, independent of a company size, adoption depends directly on training of IT and autonomous inclination towards these tools (RACERO; BUENO; GALLEGO, 2020). Finally, individual adoption and belief on the effectiveness of these software depend primarily on intrinsic factors, and is independent of previous training. Even though, training on OSS notoriously escalates those users inclined for it (RACERO; BUENO; GALLEGO, 2021).

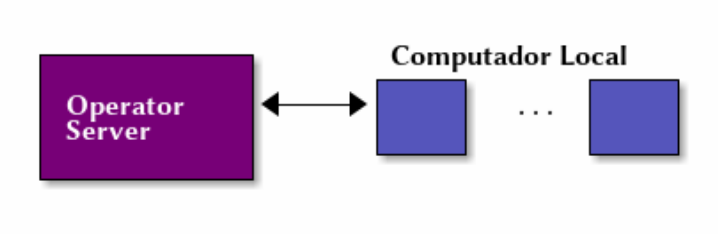
2.4 Industry Application Demonstration

2.4.1 Freqtrade

The *Freqtrade* software contains the essential pieces for a working-application of automated trading of crypto-currencies. By means of APIs, it's possible to extend the application so to communicate with the Telegram app. The program has been written in Python and the code base is open, which rely on one hundred and forty seven contributors. It's existence dates back to may 2017 (FANG et al., 2020).

The functioning of automated trading goes through a two-way communication between the local computer and a platform-server of trading - which would be Angles to a broker.

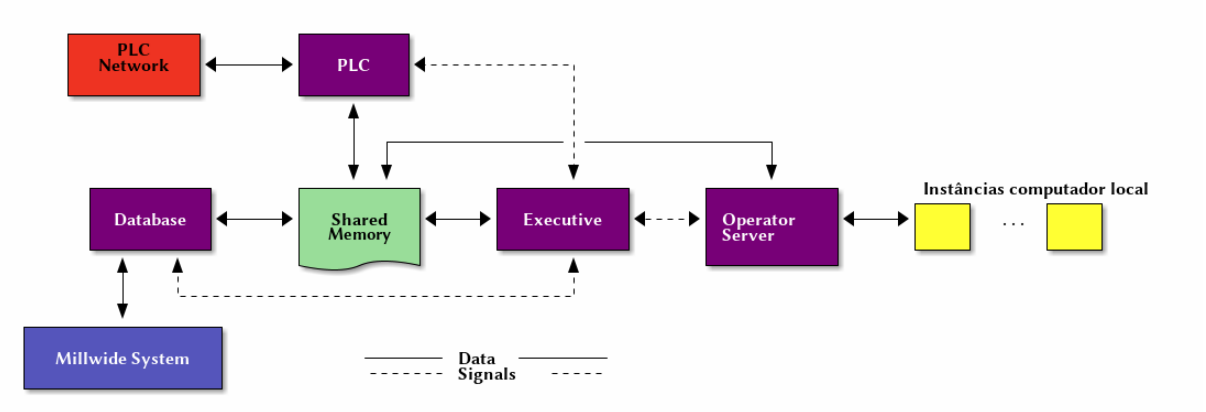
Figure 6 – a simplified schema of client-server communication



Resource: The authors.

A great part of the work on writing an automated robot, for any application end, boils down to programming communication protocols with server. Because, the robot should be capable of telling the server which operations should happen, both as means of a response to the client - e.g., a "store data on local machine" request -, as well as internal operations that should take place on the server - e.g., execute buy and sell orders on crypto-current exchanges.

Figure 7 – Schematic representation of client-server communication.



Resource: the author.

Thus, a schematic representation of this complexity could be seen in Figure 7.

By utilizing OSS, all this complexity (Figure 7) becomes close to the first schema (Figure 6). Because, all the structural abstraction has already been dealt of, by this application. To make use of this advantage, one just need to read and comprehend the documentation and extend the application to it's use.

2.4.2 OR-Tools

> Linear Programming is viewed as a revolutionary development giving us the ability for the first time to state general objectives and to find, by means of the simplex method, optimal policy decisions for a broad class of practical decision problems of great complexity. (DANTZIG, 1983)

The OR-Tools project translates into a series of code libraries developed in a open sourced manner, initially developed by Google. Possible ports exist to C++, Python, Java, C# and .Net.

The name is given due to Operational Researches Tools acronym, ORT.

We list all category of tools which are under OR-Tools [3],

1. A constraint programming solver;
2. A linear programming solver;
3. Wrappers around commercial and other open source solvers, including mixed integer solvers;
4. Bin packing and knapsack algorithms;
5. Algorithms for the Traveling Salesman Problem and Vehicle Routing Problem;
6. Graph algorithms (shortest paths, min cost flow, max flow, linear sum assignment).

The use of this tool can help solve the common problem of Scheduling. Structurally, this class of algorithms relate to solving puzzles as Sudoku - for which there are dynamical constraints upon interaction (SIMONIS, 2005).

In general, these problem fall under a mathematical and computational field called Constrained Optimization (BERTSEKAS, 2014).

Not rarely, the typical problem in Constrained Optimization can fall under a NP-hard problem (a problem for which the Combinatorics of the system have a exponential ratio, not describable by a polynomial). A classical example is the "job shop problem", for which a seemly simple and practically ubiquitous task, actually is a NP-hard problem (ZHANG et al., 2019).

Just as in non-deterministic differential equations, or in differential equations which do not have an analytical solution, it's common to use numerical solutions. When there one find a NP-hard problem, analogously, it's used genetic-algorithms and other reinforced-learning algorithms (ZHANG et al., 2019).

P vs NP problem is one of the open problems in mathematics which has deep implications to computing. Furthermore, it's listed among the ten Millennium Prize Problems, of which only one has been solved (COOK, 2006).

The Millenium Prize Problems:

[3] https://github.com/google/or-tools

- Birch and Swinnerton-Dyer conjecture
- Hodge conjecture
- Navier–Stokes existence and smoothness
- P versus NP problem
- Poincaré conjecture (solved)
- Riemann hypothesis
- Yang–Mills existence and mass gap

2.4.2.1 Application on Economics

The job scheduling problem thus is central to many areas of science. For example, generalizing the problem of Operational Research in economics, we have

> The problem of allocating scarce resources among competing ends is central to economic analysis.
>
> (...) The problem of optimal allocation of scarce resources can thus be summarized as the optimization of some objective(s) subject to constraints. (LUPTACIK et al., 2010)

Which means that this field has great application for tools as OR-Tools.

> The emphasis is given to the exposition of mathematical optimization as an instrument for qualitative analysis and to a wide range of applications in economics, including efficiency analysis, industrial economics (with focus on regulatory economics), international economics, input–output economics, quantitative economic policy and environmental economics. (LUPTACIK et al., 2010)

2.4.2.2 Application to environmental policies

Policies annually are pitched by Intergovernmental Panel on Climate Change (IPCC) on how business should constrain their business matrix of energy etc, so to meet environmental goals (KEYSSER; LENZEN, 2021).

There is an open field to transform these measures into actionable plans to enterprises. If sufficiently quantitative, such measures could be incorporated in programs is Operational Researches software (LUPTACIK et al., 2010).

There are some papers which gives Optimization Models for air pollution control (ABRAMS, 1975), countries comparative eco-efficiency (DYCKHOFF; ALLEN, 2001) etc. We cite a quotation from "Eco-efficiency analysis of industrial system in China: A data envelopment analysis approach" (ZHANG et al., 2008).

> In this connection, strategies for optimizing the use of resources or environment as expressed in more efficient way play a particularly important role. Eco-efficiency, which is an instrument for sustainability analysis,

> indicating an empirical relation in economic activities between environmental cost or value and environmental impact, has been proposed as a route to promote such a transformation (MICKWITZ et al., 2006).

And we can see that this is the case-use, at least in quantitative researches in ecology; we look into one result from US eco-efficiency research on a decade of data,

> Although an increase in both the mean RR (0.3–0.4) and the mean E-RR (0.38–0.52) scores was recorded from 1995 to 2014, this was concluded to be unsatisfactory, since the majority of the industries' eco-efficiency results were still below 0.5. (EZICI; EĞILMEZ; GEDIK, 2020)

And, the conclusion is that,

> The findings of this study suggest that substantial policy changes are required immediately to shift the negative trend in renewable energy use to comply with UN Sustainability Development Goals 7 and 13. (EZICI; EĞILMEZ; GEDIK, 2020)

We also emphasize, that this problem can seem unrelated to other things like grammar analysis on authorship, or ecological policies formulation. But, indeed they are related.

Research using eco-efficient measures upon constrains for making policies has matured. And plans utilizing it as an constrain can be pathways to saving the world from ecological decimation.

> Urban-industrial symbiosis (UIS) is an important system innovation via sectors integration, and has been widely recognized as a novel pathway for achieving regional eco-industrial development. Eco-efficiency, as a mature approach and indicator, offers an effective tool to uncover both the status and trends of such a transformation (BIAN et al., 2020).

2.4.2.3 Application to other areas

The areas that such optimization constrains are applied are uncountable. We cite some not so intuitive areas:

- Natural Language analysis and evolution (POTTS et al., 2010; HEINZ, 2018)
- Plans to maintain production of rural areas in Brazil, with no need for further deforestation (COSTA et al., 2018).
- Planning upon corporate social performance (CHEN; DELMAS, 2011; JACOBS; KRAUDE; NARAYANAN, 2016).
- Public transportation (SCHIEWE et al., 2020)
- Traffic Control (MONACHE et al., 2019)

2.5 Demonstration in Academic Applications

2.5.1 DifferentialEquations.jl

The numerical library, DifferentialEquations, has one of the best performances among numerical software (RACKAUCKAS; NIE, 2017b). The performance is comparable to implementations in FORTRAN and C. Also, this library has many ports [4], some of the languages in which you can use it are in Python, R and Julia itself.

Among the many different category of problems this library can deal with are (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),

- Discrete equations (function maps, discrete stochastic (Gillespie/Markov) simulations)
- Ordinary differential equations (ODEs)
- Split and Partitioned ODEs (Symplectic integrators, IMEX Methods)
- Stochastic ordinary differential equations (SODEs or SDEs)
- Stochastic differential-algebraic equations (SDAEs)
- Random differential equations (RODEs or RDEs)
- Differential algebraic equations (DAEs)
- Delay differential equations (DDEs)
- Neutral, retarded, and algebraic delay differential equations (NDDEs, RDDEs, and DDAEs)
- Stochastic delay differential equations (SDDEs)
- Experimental support for stochastic neutral, retarded, and algebraic delay differential equations (SNDDEs, SRDDEs, and SDDAEs)
- Mixed discrete and continuous equations (Hybrid Equations, Jump Diffusions)
- (Stochastic) partial differential equations ((S)PDEs) (with both finite difference and finite element methods)

2.5.1.1 Julia portability

The steps to use the library in Julia are,

```
# Using the package Pkg, so to manage packages.
using Pkg
# Add and/or download the DifferentialEquations.jl package in the project
Pkg.add("DifferentialEquations")
# Access all the functions defined in the package
```

[4] Capability to interoperate besides the language it was written in.

```
using DifferentialEquations
```

2.5.1.2 Python portability

In a terminal using the package manager *pip*,

```
pip install diffeqpy
```

In a terminal, running the Python interpreter,

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

An optional step is to add numba, so to optimize the code performance,

```
pip install numba
```

Finally, import the package in a file and use in a program.

```
import diffeqpy
```

2.5.1.3 R portability

To install, use

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

In a first call,

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

This way, the package will be downloaded, directly from DifferentialEquations.jl. An alternative is to use the CRAN maintained subset.

2.5.2 LaTeX

The LaTeX has fundamentally differentiate the many tasks in the typography of a document. The language permits a separation of formatting-tasks, and the content-writing. This way, the user can focus exclusively on content, in a fundamental step in document writing. And in another moment, focus solemnly on format and aesthetics.

Therefore, there a quality gain in production. Just as one is rewarded with the total control of the document, because every graphical disposition and behavior is defined by

open code. Furthermore, one can extend already written templates and packages which do usual formatting. The typographical system in LaTeX has been considered, also, one of the most sophisticated software of it's category, mostly because of this bottom-up paradigm one naturally uses (HARALAMBOUS, 2007).

The LaTeX, technically, is the union between the typographical language TeX, invented by Donald Knuth, for high quality documents (KNUTH, 1986). And, the powerful macro ecosystem, which extend the TeX programming, for which we call LaTeX.

Initially LaTeXwas developed by Leslie Lamport, with his personal use-case formatting programs (LAMPORT, 1994). Thus, LaTeX is not just a language, but a conjunction of macros and standard environments, which is actively maintained, modified, and extended by the open community.

2.5.2.1 The canonical ABNT class for scientific production

What is called as canonical are a set of documents following the more general and robust ABNT (Associação Brasileira de Normas Técnicas) norms. These canonical forms exist for scientific articles, technical reports, academic works as thesis, dissertations and research projects, books and presentations (ABNTEX, 2012).

> 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)

The norms prescribed by the canonical model are:

- **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 secções 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.

2.6 Canonical works on Scientific Computer Science

2.6.1 Structure and Interpretation of Classical Mechanics (SCIM)

The scientific library of classical mechanics, written in Scheme (a Lisp dialect), has been used to teach master's degree MIT students. Accompanied to the library there is the book, which servers as didactic material for the course (SUSSMAN; WISDOM, 2015).

The library and the book are open sourced.

2.6.2 Structure and interpretation of computer programs (SICP)

The scheme course (SICP), which is teach as the foundations of computer science course on MIT, has accompanied as textbook one of the most world wide recognized books in computer history (ABELSON; SUSSMAN, 1996).

This course has been the pioneer on givin emphasis to the epistemological activity, when programming. The use of Scheme, which has a uniform notation (only one form), was revolutionary. This has shown to work well, instead of presenting the trending language in Industry, in a given moment - which is the standard approach.

The SCIM (subsection 2.6.1) course as a predecessor to SICP. The course and the text book still are in use in MIT and a number of other universities around the world. A non-exaustive list can be found at <https://mitpress.mit.edu/sites/default/files/sicp/adopt-list.html>. At least twenty universities adopts, distributed in more than eight countries.

These materials, textbook and the MIT classes, are both free and open sourced.

2.6.3 SCIMUtils - (SCIM) Portability in Clojure

These are dedicated libraries for rewriting the SCIM programs in Clojure, which ports many functionalities mentioned in the scientific library SCIMThe library and the course's programs are not the same thing (SMITH; RITCHIE, 2016). This library, therefore, has been ported to Clojure, which is the most industry-used Lisp modern-dialect. For example, Nubank bought Cognitect which was the first and most successful application written in Clojure (HAMILTON, 2020).

The language has a double quality in it's program. Due to the language supporting multiple definitions of a function under a name (polymorphism), it's possible to very easily switch and integrate numerical and symbolical computations.

Changing inputs one would have one or the other output, by inference of the Clojure compiler. This is particularly interesting when solving, for example, Lagrange equations of motion. One can have both the symbolical results outputted in TEX, and by a change in inputs, in the same procedure, have numerical data which can be used to simulate the behavior of physical systems.

3 Materials and Methods

The planning of the monograph has followed the steps:

1. Contact and develop strategies with the supervisor;
2. Deliniating the main theme;
3. Given the date constrains, schedule an fitting agenda;
4. Outline the structure of the TCC;
5. Gather of exploratory notes about the topic;
6. Biliographical reseach;
7. Writting each dedicated topic;
8. Reuse past knowledge and use personal practical examples;
9. Continuous revision, for each step of the monograph;
10. Organize the presentation;

3.1 Initial Strategy

The first steps taken were to determine what would be the best match of student-advisor, based on mutual interest on the topic. Then, began the first general schema of the work, and the schedules.

3.2 Theme delineation

Given that the theme OSS on Industry and Academia has indefinetly many branches to discuss, we urged to narrow down the boudaries of the topic to those applications of most familiity to the student.

In the continuous reviews, we further narrowed the topic by wheigthing if deepening the analyses wouldn't deviate from the main topics.

3.3 Cronological Schedule

We followed the Gantt diagram method to orient ourselves, which is a visual aid to help on following deadlines. At the same time, the diagram helps to situate oneself in relation to what should the team be doing.

3.4 The monograph theme

As this monograph mainly falls under an Exploratory Analysis, the main focus was the theme's literature. We show programs developed by the open community.

On the Academic Application's branch, the emphasis was given to auxiliary software for research. Also, we shown some software that may as well be used for teaching physical concepts.

In the short experience of the student on Industry, we found examples of software he already made use of.

3.5 Gathering various notes

One of the open sourced software of most use to the student is org-mode. In *.org* files, one can take notes, just as in note applications found in any device. A lot of material was gathered from those notes to compose the results.

3.6 Bibliographical Research

The bibliographical review intended to facilitate the comprehension of Software Industry. We gathered some cuts of the history of FOSS. Furthermore, we brought information about some open software which are considered state of the art, or iconic, on their field of application.

Most of the historical content about the FOSS movement and software in general can be found in websites. Particularly, GNU was the main source of FOSS history. In contrast, most sources on software development and comparative functionality was found on scientific articles.

We used Google Scholar and Association for Computing Machinery's (ACM) platform. The student had access due to being a member to the research engine and papers they have. The content of ACM is mainly about software and theoretical computation in general.

3.7 Continuous Review

A fundamental part of the research's organization conciseness was due to the frequent interaction between the student and the advisor.

3.8 Presentation Organization

The use of visual software coupled with org-mode [1] makes an presentation interactive. So, much of the work done in the monograph was reused for presentation.

[1] There are dedicated packages in Emacs to make presentations with org-mode files https://github.com/howardabrams/demo-it

4 Results and Discussion

4.1 Advisor expertise and the theme

One of the applications the student supposed could be done with OSS was with Lagrangians. SCIMUtils (subsection 2.6.3) exist in Clojure - the port of the famous work SCIM. and the advisor had given lectures on the subject of Classical Mechanics on the university courses. At the same time, the advisor itself is an user-developer, using Linux. So, the themes and expertise matched.

4.2 Theme delineation

The GNU and the OSS initiative have indefinitely many subjects that could be tackled. There are virtually FOSS written for application on any human enterprise.

We opted to work on applications most familiar to the student. Also, we the work focused on presenting the most significant aspects of FOSS, in geral, for Academia and Industry.

Thus, the delimitation on showing the framework for an working FOSS OS. The delimitation to the OR-Tools and Freqtrade in Industry. Finally, the use of Clojure, Julia, Python and R as means to present graphical and numerical applications to abstract ideas in mathematics.

4.2.1 Industry Applications

4.2.1.1 OR-Tools

The language used to write this application is C++ [1]. But, there are ports to Python, Java (thus, Clojure), .Net. All languages have this program licensed in *copy-left* terms.

OR-Tools has been designed to tackle conspicuous problems: one of them, scheduling. Virtually any social-economic enterprise use scheduling in some moment.

4.2.1.1.1 Scheduling work shifts problem

We offer an example in the following setting. A hospital needs to create a schedule for four nurses over a three-day period, subject to the following conditions:

1. Each day is divided into three 8-hour shifts.

[1] https://github.com/google/or-tools

2. Every day, each shift is assigned to a single nurse, and no nurse works more than one shift.
3. Each nurse is assigned to at least two shifts during the three-day period.
4. There are 10 nurses.
5. There are 7 days of work.

In this problem, the program would translate to [2], Importing the OR-Tools utilities,

```
from ortools.sat.python import cp_model
```

Stating the constants, and standard input/output functions,

```
class NursesPartialSolutionPrinter(cp_model.CpSolverSolutionCallback):
    """Print intermediate solutions."""

    def __init__(self, shifts, num_nurses, num_days, num_shifts, sols):
        cp_model.CpSolverSolutionCallback.__init__(self)
        self._shifts = shifts
        self._num_nurses = num_nurses
        self._num_days = num_days
        self._num_shifts = num_shifts
        self._solutions = set(sols)
        self._solution_count = 0

    def on_solution_callback(self):
        if self._solution_count in self._solutions:
            print('Solution %i' % self._solution_count)
            for d in range(self._num_days):
                print('Day %i' % d)
                for n in range(self._num_nurses):
                    is_working = False
                    for s in range(self._num_shifts):
                        if self.Value(self._shifts[(n, d, s)]):
                            is_working = True
                            print('  Nurse %i works shift %i' % (n, s))
                    if not is_working:
                        print('  Nurse {} does not work'.format(n))
            print()
        self._solution_count += 1

    def solution_count(self):
        return self._solution_count
```

Giving fixed values to the program,

```
def main():
```

[2] This problem is a variation from the one on the project's documentation, https://developers.google.com/optimization/scheduling/employee_scheduling

```
# Data.
num_nurses = 10
num_shifts = 3
num_days = 7
all_nurses = range(num_nurses)
all_shifts = range(num_shifts)
all_days = range(num_days)
```

Creating the model constraints,

```
# Creates the model.
model = cp_model.CpModel()

# Creates shift variables.
# shifts[(n, d, s)]: nurse 'n' works shift 's' on day 'd'.
shifts = {}
for n in all_nurses:
    for d in all_days:
        for s in all_shifts:
            shifts[(n, d,
                    s)] = model.NewBoolVar('shift_n%id%is%i' % (n, d, s
                                                                ))

# Each shift is assigned to exactly one nurse in the schedule period.
for d in all_days:
    for s in all_shifts:
        model.Add(sum(shifts[(n, d, s)] for n in all_nurses) == 1)

# Each nurse works at most one shift per day.
for n in all_nurses:
    for d in all_days:
        model.Add(sum(shifts[(n, d, s)] for s in all_shifts) <= 1)

# Try to distribute the shifts evenly, so that each nurse works
# min_shifts_per_nurse shifts. If this is not possible, because the
                                        total
# number of shifts is not divisible by the number of nurses, some
                                        nurses will
# be assigned one more shift.
min_shifts_per_nurse = (num_shifts * num_days) // num_nurses
if num_shifts * num_days % num_nurses == 0:
    max_shifts_per_nurse = min_shifts_per_nurse
else:
    max_shifts_per_nurse = min_shifts_per_nurse + 1
for n in all_nurses:
    num_shifts_worked = 0
    for d in all_days:
        for s in all_shifts:
```

```
                num_shifts_worked += shifts[(n, d, s)]
        model.Add(min_shifts_per_nurse <= num_shifts_worked)
        model.Add(num_shifts_worked <= max_shifts_per_nurse)
```

Call the $\text{cp}_model to transform the constrains in a solvable format$,

```
    # Creates the solver and solve.
    solver = cp_model.CpSolver()
    solver.parameters.linearization_level = 0
```

Finally, print a few of the solutions (5) and give some feedback on performance and analytics of the problem.

```
    # Display the first five solutions.
    a_few_solutions = range(5)
    solution_printer = NursesPartialSolutionPrinter(shifts, num_nurses,
                                                    num_days, num_shifts,
                                                    a_few_solutions)
    solver.SearchForAllSolutions(model, solution_printer)

    # Statistics.
    print()
    print('Statistics')
    print('  - conflicts       : %i' % solver.NumConflicts())
    print('  - branches        : %i' % solver.NumBranches())
    print('  - wall time       : %f s' % solver.WallTime())
    print('  - solutions found : %i' % solution_printer.solution_count())

if __name__ == '__main__':
    main()
```

The result:

```
Solution 4
Day 0
  Nurse 0 does not work
  Nurse 1 does not work
  Nurse 2 does not work
  Nurse 3 does not work
  Nurse 4 does not work
  Nurse 5 does not work
  Nurse 6 does not work
  Nurse 7 works shift 0
  Nurse 8 works shift 1
  Nurse 9 works shift 2
Day 1
  Nurse 0 does not work
  Nurse 1 does not work
```

```
  Nurse 2 does not work
  Nurse 3 does not work
  Nurse 4 does not work
  Nurse 5 does not work
  Nurse 6 does not work
  Nurse 7 works shift 2
  Nurse 8 works shift 1
  Nurse 9 works shift 0
(...)
Day 6
  Nurse 0 works shift 1
  Nurse 1 works shift 0
  Nurse 2 works shift 2
  Nurse 3 does not work
  Nurse 4 does not work
  Nurse 5 does not work
  Nurse 6 does not work
  Nurse 7 does not work
  Nurse 8 does not work
  Nurse 9 does not work

Statistics
  - conflicts       : 3540
  - branches        : 67968438
  - wall time       : 102.701620 s
  - solutions found : 561425
```

When there are too many solutions to a constrain problem, that means there are too much resources to the given constrain. That is, the hospital could operate with less workers [3]. Let's use 3 nurses, keeping the other constrains.

This gives us, on termination,

```
Statistics
  - conflicts       : 386
  - branches        : 2683080
  - wall time       : 15.481512 s
  - solutions found : 279936
```

But, if we then change the constrains to number of shifts being 4 - that is, 6 hour-work.

```
Statistics
  - conflicts       : 10
  - branches        : 194
  - wall time       : 0.002517 s
  - solutions found : 0
```

[3] We stopped the program before it found all solutions.

Of course, because one of the nurses would need to work more than once a day. This would break one of our constrained-condition. We note, though, that there is no continuity in-between the possibility of no solution and the most optimized one.

Furthermore, the solutions have exponential growth on the **event space**. Thus, not being a polynomial run-time model, we call this problem a Nondeterministic-Polynomial time (NP) on decision.

4.2.1.2 Freqtrade

Freqtrade automate crypto-trading. The software use is pretty straight forward, once one has developed an strategy, in Python.

The strategy using Boiling Bands (BB) and Relative Strength Index (RSI) can be understood in words as the principle: buy when prices are oversold and sell when it's on an average price[4].

Downloading our repository with FreqTrade configurations,

```
git clone https://github.com/BuddhiLW/studious-carnival.git
```

Navigate to strategies folder files,

```
cd studious-carnival/user_data/strategies
```

Where the strategy-06-bb-rsi.py will have the strategy written in Python. One of the most important parts is the **definition of the indicators used (BB and RSI)** [5].

```
    def populate_indicators(self, dataframe: DataFrame, metadata: dict) ->
                                            DataFrame:


        # RSI
        dataframe['rsi'] = ta.RSI(dataframe)

        # Bollinger Bands
        bollinger1 = qtpylib.bollinger_bands(qtpylib.typical_price(
                                                dataframe), window=20, stds=1
                                                )
        dataframe['bb_lowerband1'] = bollinger1['lower']
        dataframe['bb_middleband1'] = bollinger1['mid']
        dataframe['bb_upperband1'] = bollinger1['upper']

        bollinger3 = qtpylib.bollinger_bands(qtpylib.typical_price(
                                                dataframe), w
```

[4] Using the repository with (already) optimized bb-rsi strategies
[5] Notes on terms: σ means standard deviation.

```
indow=30, stds=3)
        dataframe['bb_lowerband3'] = bollinger3['lower']
        dataframe['bb_middleband3'] = bollinger3['mid']
        dataframe['bb_upperband3'] = bollinger3['upper']

        return dataframe
```

Then, we use these to automate our buys and sells conditions. First, coding when to buy: "if the current price is bellow three σ of the mean."

```
    def populate_buy_trend(self, dataframe: DataFrame, metadata: dict) ->
                                            DataFrame:
        dataframe.loc[
            (
                # (dataframe['rsi'] > 47) &
                (dataframe["close"] < dataframe['bb_lowerband3'])
            ),
            'buy'] = 1

        return dataframe
```

The sell strategy: "sell when RSI is above 97 and above one σ bellow mean."

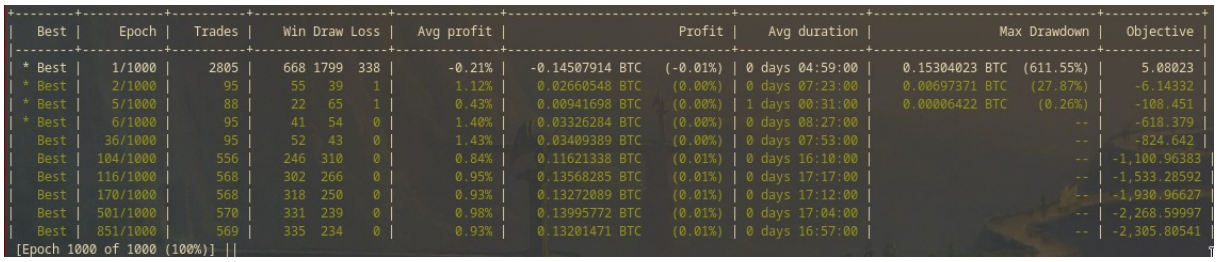
```
    def populate_sell_trend(self, dataframe: DataFrame, metadata: dict) ->
                                            DataFrame:
        dataframe.loc[
            (
                (dataframe['rsi'] > 97) &
                (dataframe["close"] > dataframe['bb_middleband1'])
            ),
            'sell'] = 1

        return dataframe
```

These seemly arbitrary values for the strategy came from the hyper-optimization algorithm, builtin the program. We can use this functionality by the following command:

```
sudo docker-compose run --rm freqtrade hyperopt -s bbrsi6 --hyperopt
                                        bbrsiopt --hyperopt-loss
                                        SortinoHyperOptLossDaily -i 1h -j 8 -
                                        e 2000 --timerange 20210101-
```

This means command means the following: "choose strategy bbrsi6 (-s), with bbrsiopt parameters of optimization (–hyperopt), the constrain on optimization being SortinoHyperOptLossDaily (–hyperopt-loss), use trade timeframe of 1h (-i), use computing power of 8 cores (-j 8), optimize through 2000 iterations (-e), and use the data since the beginning of 2021 until today (–timerange)".

Figure 8 – Epochs of strategy optimization



	Best	Epoch	Trades	Win Draw Loss	Avg profit	Profit	Avg duration	Max Drawdown	Objective
	* Best	1/1000	2805	668 1799 338	-0.21%	-0.14507914 BTC (-0.01%)	0 days 04:59:00	0.15304023 BTC (611.55%)	5.08023
	* Best	2/1000	95	55 39 1	1.12%	0.02660548 BTC (0.00%)	0 days 07:23:00	0.00697371 BTC (27.87%)	-6.14332
	* Best	5/1000	88	22 65 1	0.43%	0.00941698 BTC (0.00%)	1 days 00:31:00	0.00006422 BTC (0.26%)	-108.451
	* Best	6/1000	95	41 54 0	1.40%	0.03326284 BTC (0.00%)	0 days 08:27:00	--	-618.379
	Best	36/1000	95	52 43 0	1.43%	0.03409389 BTC (0.00%)	0 days 07:53:00	--	-824.642
	Best	104/1000	556	246 310 0	0.84%	0.11621338 BTC (0.01%)	0 days 16:10:00	--	-1,100.96383
	Best	116/1000	568	302 266 0	0.95%	0.13568285 BTC (0.01%)	0 days 17:17:00	--	-1,533.28592
	Best	170/1000	568	318 250 0	0.93%	0.13272089 BTC (0.01%)	0 days 17:12:00	--	-1,930.96627
	Best	501/1000	570	331 239 0	0.98%	0.13995772 BTC (0.01%)	0 days 17:04:00	--	-2,268.59997
	Best	851/1000	569	335 234 0	0.93%	0.13201471 BTC (0.01%)	0 days 16:57:00	--	-2,305.80541

[Epoch 1000 of 1000 (100%)] ||

Optimization application running. Resource: The authors.

Figure 9 – Optimized Strategy Suggestion

```
Best result:

  851/1000:    569 trades. 335/234/0 Wins/Draws/Losses. Avg profit   0.93%. Median profit   1.03%. Total profit  0.13201471 BTC (   0.01%). Avg duration 16:57:00 min. Objective: -2305.80541


  # Buy hyperspace params:
  buy_params = {
      "rsi-value": 7,
      "rsi-enabled": True,
      "trigger": "bb_lower3",
  }

  # Sell hyperspace params:
  sell_params = {
      "sell-rsi-value": 93,
      "sell-rsi-enabled": True,
      "sell-trigger": "sell-bb_lower1",
  }

  # ROI table:
  minimal_roi = {
      "0": 0.21906,
      "37": 0.02544,
      "84": 0.01032,
      "188": 0
  }

  # Stoploss:
  stoploss = -0.35101

  # Trailing stop:
  trailing_stop = True  # value loaded from strategy
  trailing_stop_positive = 0.27614  # value loaded from strategy
  trailing_stop_positive_offset = 0.33464  # value loaded from strategy
  trailing_only_offset_is_reached = True  # value loaded from strategy
```

Output to be placed under the strategy file. Resource: The authors.

By configuring the file,

```
cd studious-carnival/user_data/config.json
```

We can hook the application to our Telegram account and our exchange. For example, one of the options of exchange is Binance [6].

```
"exchange": {
        "name": "binance",
        "key": "<<public-key>>",
        "secret": "<<secret-key>>",
        "ccxt_config": {"enableRateLimit": true},
        "ccxt_async_config": {
            "enableRateLimit": true,
            "rateLimit": 200
        },
```

[6] The markdown "«»" symbolizes that is a key you would need to provide

And, the Telegram configuration can be configured as,

```
"telegram": {
        "enabled": true,
        "token": "<<token>>",
        "chat_id": "1272433706"
      },
```

Also, it's possible to test our strategy on past data. We have to call the "backtesting" command to that end.

```
sudo docker-compose run --rm freqtrade backtesting --strategy bbrsi6 --
                                                timeframe 1h
```

Figure 10 – Output backtesting history

```
========== BACKTESTING REPORT ==========
|      Pair |   Buys |   Avg Profit % |   Cum Profit % |   Tot Profit BTC |   Tot Profit % |     Avg Duration |   Win  Draw  Loss  Win% |
|-----------+--------+----------------+----------------+------------------+----------------+------------------+-------------------------|
|   VET/BTC |    105 |           1.01 |         106.16 |       0.02656635 |           0.00 |         21:41:00 |    48    57     0   100 |
|  DOGE/BTC |     86 |           1.04 |          89.20 |       0.02232238 |           0.00 |  4 days, 7:28:00 |    33    52     1  38.4 |
|   SOL/BTC |     57 |           0.83 |          47.42 |       0.01186661 |           0.00 |         8:22:00 |    30    27     0   100 |
|  ATOM/BTC |     68 |           0.68 |          46.45 |       0.01162405 |           0.00 |         2:30:00 |    40    28     0   100 |
|   ADA/BTC |     86 |           0.48 |          41.24 |       0.01032058 |           0.00 |        19:34:00 |    40    46     0   100 |
|   DOT/BTC |     31 |           1.03 |          31.79 |       0.00795654 |           0.00 |         1:43:00 |    22     9     0   100 |
|   LTC/BTC |     98 |           0.32 |          31.70 |       0.00793260 |           0.00 |         4:32:00 |    45    53     0   100 |
|  DASH/BTC |     93 |           0.34 |          31.37 |       0.00785043 |           0.00 |         9:01:00 |    39    54     0   100 |
|   SNX/BTC |     33 |           0.90 |          29.74 |       0.00744181 |           0.00 |        10:04:00 |    23    10     0   100 |
|   ONT/BTC |     83 |           0.32 |          26.52 |       0.00663698 |           0.00 |         9:19:00 |    42    41     0   100 |
|  AAVE/BTC |     24 |           1.09 |          26.04 |       0.00651708 |           0.00 |         7:48:00 |    14    10     0   100 |
|   ETH/BTC |     97 |           0.25 |          24.11 |       0.00603262 |           0.00 |        16:09:00 |    40    57     0   100 |
|   BCH/BTC |     79 |           0.28 |          21.86 |       0.00546967 |           0.00 |        14:27:00 |    36    43     0   100 |
|   UNI/BTC |     23 |           0.64 |          14.81 |       0.00370545 |           0.00 |        11:08:00 |    11    12     0   100 |
|   MKR/BTC |     28 |           0.47 |          13.30 |       0.00332745 |           0.00 |        20:09:00 |    14    14     0   100 |
|  EGLD/BTC |     28 |           0.40 |          11.25 |       0.00281612 |           0.00 |        17:21:00 |    16    11     1  57.1 |
|  LINK/BTC |     68 |           0.15 |          10.39 |       0.00260063 |           0.00 |  2 days, 4:39:00 |    39    28     1  57.4 |
|   XLM/BTC |     97 |           0.05 |           4.91 |       0.00122990 |           0.00 |         6:58:00 |    43    53     1  44.3 |
|   BNB/BTC |     77 |           0.04 |           3.26 |       0.00081480 |           0.00 |  2 days, 4:57:00 |    42    34     1  54.5 |
|  WBTC/BTC |     29 |           0.00 |           0.03 |       0.00000873 |           0.00 |  5 days, 2:17:00 |     1    28     0   100 |
|   XRP/BTC |     67 |          -0.01 |          -0.68 |      -0.00017112 |          -0.00 |  3 days, 1:41:00 |    37    29     1  55.2 |
|   ETC/BTC |     81 |          -0.10 |          -8.30 |      -0.00207804 |          -0.00 |   1 day, 6:57:00 |    38    42     1  46.9 |
|  AVAX/BTC |     24 |          -0.69 |         -16.66 |      -0.00417028 |          -0.00 |  2 days, 3:58:00 |    13    10     1  54.2 |
|   XMR/BTC |     63 |          -0.38 |         -23.68 |      -0.00592588 |          -0.00 |  2 days, 3:32:00 |    26    36     1  41.3 |
|  COMP/BTC |     34 |          -0.72 |         -24.43 |      -0.00611441 |          -0.00 |        22:09:00 |    17    16     1  50.0 |
|   FIL/BTC |     29 |          -0.95 |         -27.61 |      -0.00690888 |          -0.00 |        16:48:00 |    14    14     1  48.3 |
|   FTT/BTC |     57 |          -0.49 |         -27.98 |      -0.00700263 |          -0.00 |  4 days, 2:08:00 |    21    35     1  36.8 |
|  IOTA/BTC |     63 |          -0.61 |         -38.29 |      -0.00958158 |          -0.00 | 3 days, 10:23:00 |    31    30     2  49.2 |
| SUSHI/BTC |     28 |          -1.41 |         -39.55 |      -0.00989760 |          -0.00 |         6:56:00 |    14    12     2  50.0 |
|   EOS/BTC |     58 |          -0.71 |         -41.35 |      -0.01034744 |          -0.00 | 4 days, 11:47:00 |    25    31     2  43.1 |
|     TOTAL |   2006 |           0.17 |         350.13 |       0.08761935 |           0.01 |   1 day, 8:57:00 |   946  1039    21  47.2 |
========== SELL REASON STATS ==========
|   Sell Reason |   Sells |   Win  Draws  Loss  Win% |   Avg Profit % |   Cum Profit % |   Tot Profit BTC |   Tot Profit % |
|---------------+---------+--------------------------+----------------+----------------+------------------+----------------|
|           roi |    1985 |   946  1039     0   100 |           0.58 |        1151.29 |         0.288111 |          34.89 |
|     stop_loss |      21 |     0     0    21     0 |         -38.15 |        -801.17 |        -0.200492 |         -24.28 |
========== LEFT OPEN TRADES REPORT ==========
|   Pair |   Buys |   Avg Profit % |   Cum Profit % |   Tot Profit BTC |   Tot Profit % |   Avg Duration |   Win  Draw  Loss  Win% |
|--------+--------+----------------+----------------+------------------+----------------+----------------+-------------------------|
|  TOTAL |      0 |           0.00 |           0.00 |       0.00000000 |           0.00 |           0:00 |     0     0     0     0 |
```

Performance per crypto-coin pair. Resource: The authors.

Finally, our application can be run in dry-mode (real-time simulation) [7], or with real money.

```
"dry_run": false,
```

[7] also configured under config.json

Figure 11 – Summary about the backtesting performance

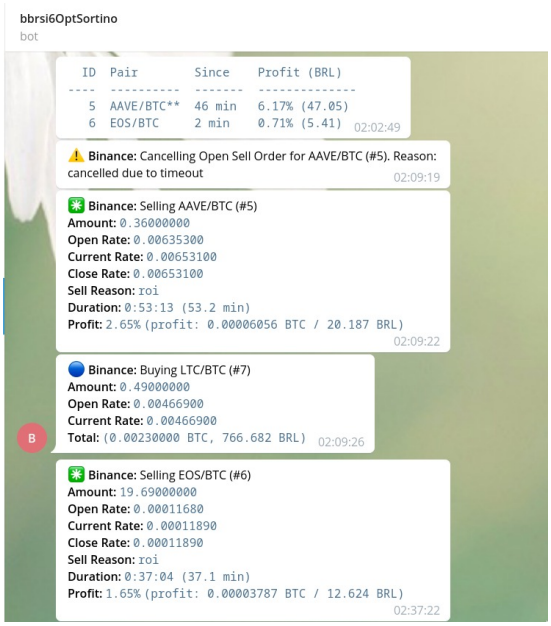
SUMMARY METRICS	
Metric	Value
Backtesting from	2019-10-02 06:00:00
Backtesting to	2021-05-08 14:00:00
Max open trades	33
Total/Daily Avg Trades	2006 / 3.43
Starting balance	1000.00000000 BTC
Final balance	1000.08761935 BTC
Absolute profit	0.08761935 BTC
Total profit %	0.01%
Avg. stake amount	0.02500000 BTC
Total trade volume	50.15000000 BTC
Best Pair	VET/BTC 106.16%
Worst Pair	EOS/BTC -41.35%
Best trade	SUSHI/BTC 18.6%
Worst trade	IOTA/BTC -38.15%
Best day	0.02088531 BTC
Worst day	-0.02834548 BTC
Days win/draw/lose	304 / 263 / 17
Avg. Duration Winners	1:00:00
Avg. Duration Loser	51 days, 5:49:00
Rejected Buy signals	0
Min balance	999.97348770 BTC
Max balance	1000.08761935 BTC
Drawdown	412.41%
Drawdown	0.10320519 BTC
Drawdown high	0.07669289 BTC
Drawdown low	-0.02651230 BTC
Drawdown Start	2020-10-06 09:00:00
Drawdown End	2021-01-01 18:00:00
Market change	300.21%

Major information. Resource: The authors.

Launch the application in the shell, with the following command.

```
sudo docker-compose run --rm freqtrade trade -s bbrsi6 -c user_data/config.
                                        json
```

Figure 12 – Telegram Interaction, application running



We can monitor and control the application through telegram. Resource: The authors.

4.2.2 Academia Applications

We opted to work with applications written in Python, Julia and Clojure(Script). Because, the student already is familiar to these languages. In fact, many of the applications brought in this work has been cuts of real problems the authors faced in Academia. These simulations are thus a deepening of how these solutions emerge.

4.2.2.1 DifferentialEquations.jl

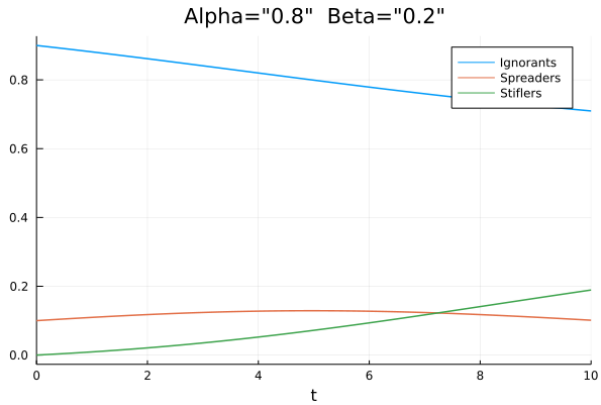
Julia has been invented to work with scientific computation. The optimized computation of Julia compares to FORTRAN and C++. Thus, we chose to use a totally julia-written package, called *DifferentialEquations.jl*. This package can be used to solve a wide range of differential equations.

4.2.2.1.1 Rumor Propagation Model

Piqueira (PIQUEIRA, 2010) proposed a model based on Daley-Kendal (DALEY; KENDALL, 1964) for rumor propagation[8] as,

$$\begin{aligned} \dot{I} &= \beta k S I \\ \dot{S} &= \beta k S I - \alpha k S (S + R) \\ \dot{R} &= \alpha k S (S + R) \end{aligned} \tag{4.1}$$

Figure 13 – Rumor Propagation Model - Hard information



$\alpha = 0.8$ and $\beta = 0.2$. Resource: The authors.

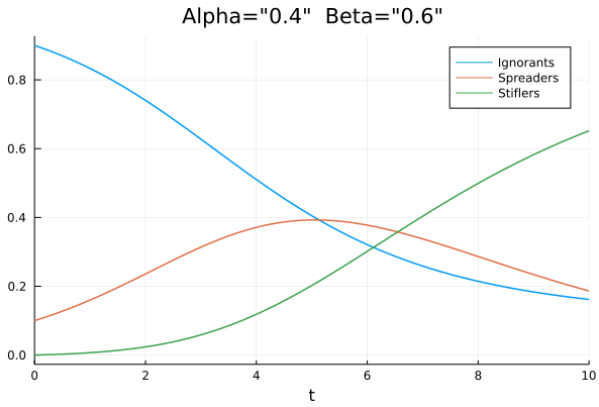
And, we could easily use Julia and it's package to investigate the behavior of these equations. By reading the paper, we understand α means how hard an information is hard to get, learn or remember etc. At the same time, β has to do with how much an

[8] There are many other versions considering some kind of variation. In another modern paper, Piqueira consider fact-checkers (PIQUEIRA; ZILBOVICIUS; BATISTELA, 2020)

information is attractive, scarce (meaning unique), socially relevant etc. And, k has to do with normalization of the α, β and the total population. if $\alpha + \beta = 1$ then we can set $k = 1$, simplifying the equations.

We then can then experiment and derive conclusions, for an example, that extremely important information which is also hard to get will have a limiting rate of Ignorants and people who contain that information, but do not propagate it (Stiflers). At the same time, spreaders will have an oscillating response, just as to maintain the limiting rate of "knowers" (fig:diffeq-case1).

Figure 14 – Rumor Propagation Model - Effects on Spreaders



$\alpha = 0.4$ and $\beta = 0.6$. Resource: The authors.

At the same time, we can learn about the counter-intuitive effect of making a relevant information, with good spreading changes, too accessible. That can make part of the public unaware of it. Because the Spreaders may lose interest in talking about it, due to too many people talking about it. And, then, all of the sudden, these Spreaders become Stiflers who forgot or do not talk about the subject anymore for whatever reason (Figure 14).

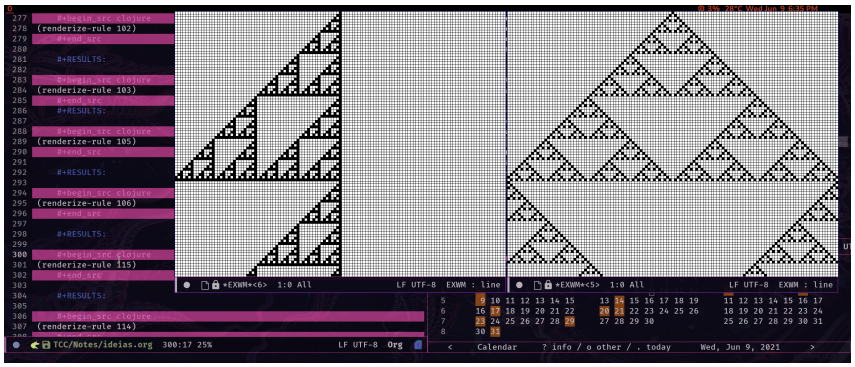
4.2.2.2 Teaching abstract concepts

Some animations in Clojure can be done using a great deal of libraries. And, some of them can be used for understanding STEM related abstractions.

4.2.2.2.1 Cellular Automata

Proposed by Wolfram (WOLFRAM, 1983), a useful way to understand and contemplate complexity can be by visualizing cellular automata. From these seemingly simple rules arise immense complexity (Figure 15).

Figure 15 – Cellular Automata with Clojure

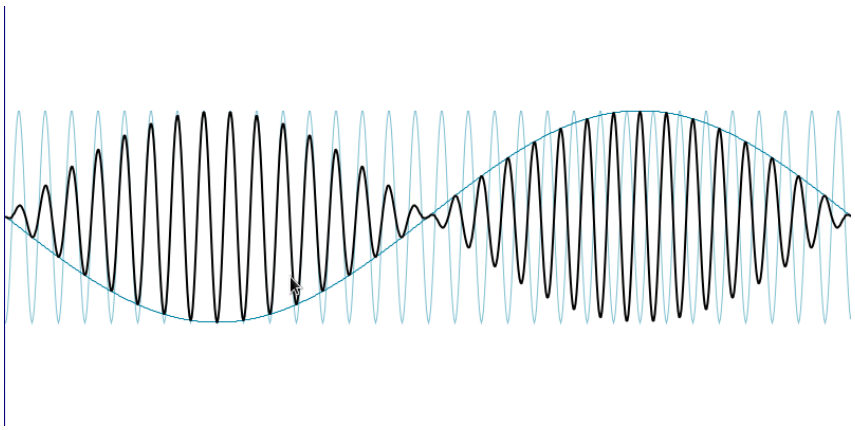


Simulating simple rules, arising complexity. Resource: The authors.

4.2.2.2.2 Electromagnetism

In Clojure2D library, we can write waves that imitate electromagnetic waves behavior (Figure 16).

Figure 16 – Electromagnetism with Clojure

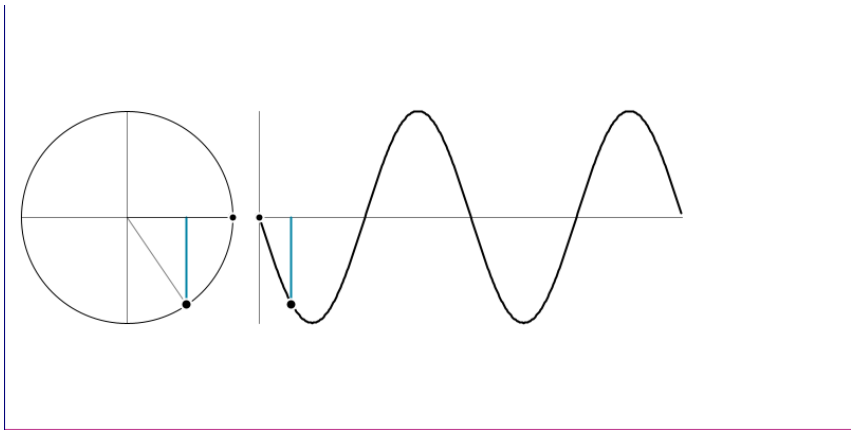


Resource: The authors.

4.2.2.2.3 Calculus

Pre-calculus knowledge of how $sin(\theta)$ behaves. Probably, explain rates of variation and other concepts with interactive simulations. The student would be able to actively engaged, manipulating the code, angular rate etc (fig:calc).

Figure 17 – Calculus and rates of changes

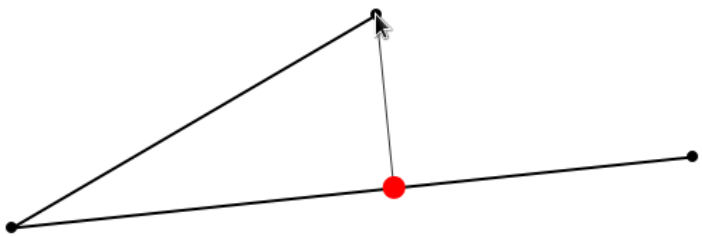


Resource: The authors.

4.2.2.2.4 Analytical Geometry

With Clojure2D there are implemented simulations of vector projection simulation. These could be used in teaching analytical geometry (Figure 18).

Figure 18 – Vector projection simulation

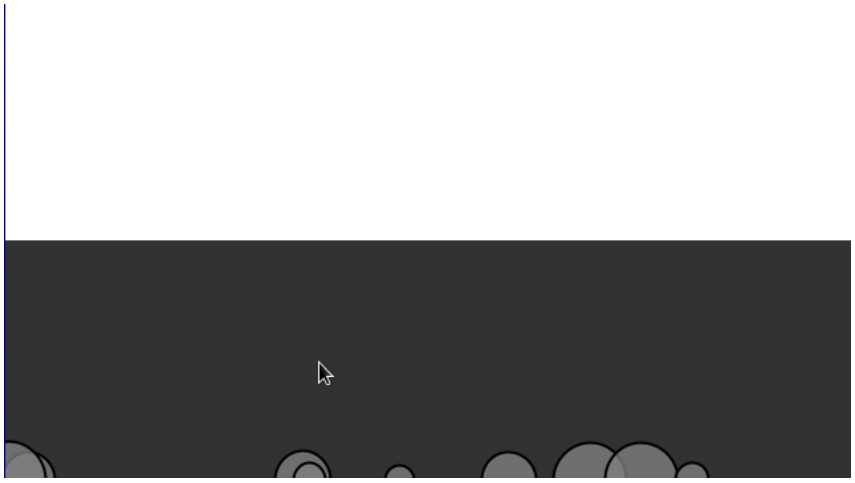


Resource: The authors.

4.2.2.2.5 Fluid Viscosity

Also, one could use implemented simulations of free fall bodies changing of medium. These models show how a body changes motion under such changes (Figure 19).

Figure 19 – Body under gravity - Viscosity

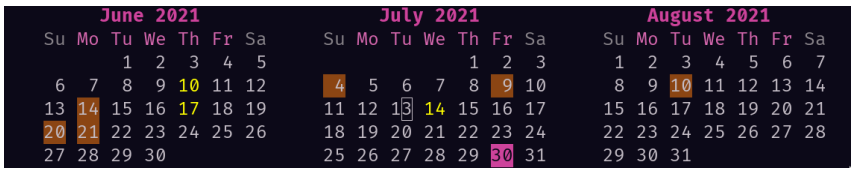


Resource: The authors.

4.3 Chronological Organization

The present work has been manager through Gantt diagrams[9]. Also, we used dynamic agendas builtin in Emacs. Thus all visual framing of time has been done by either one these two.

Figure 20 – Emacs's Calendar

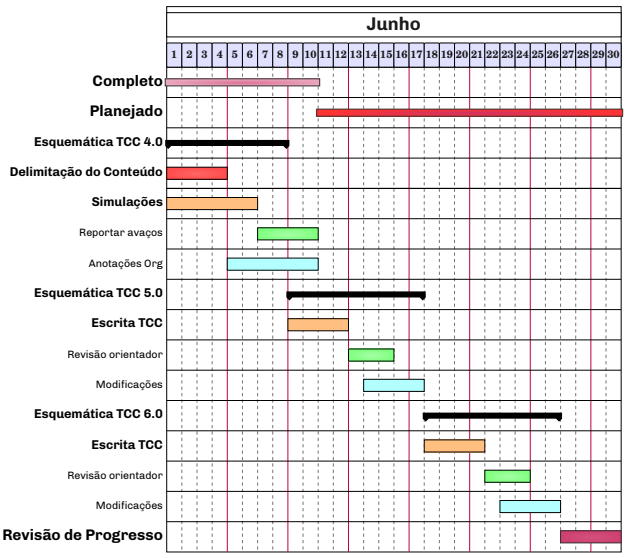


Schedules, deadlines and vacations marked. Resource: The authors.

The system TGD (To Get Done) and TODO-NEXT-DONE has been implemented using Org-mode synchronized with Org-agenda.

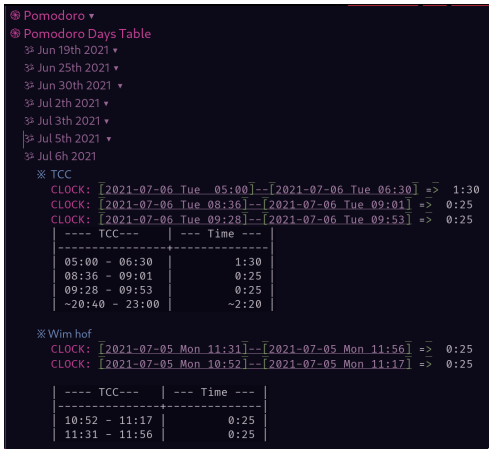
[9] Using LaTeX

Figure 21 – Gantt Chart produced with latex



Gantt for July. Resource: The authors.

Figure 22 – Daily Pomodoro timing (Using Org-pomodoro)



Tables written in Metrics.org file. Resource: The authors.

4.4 Note collection and ABNTeX2

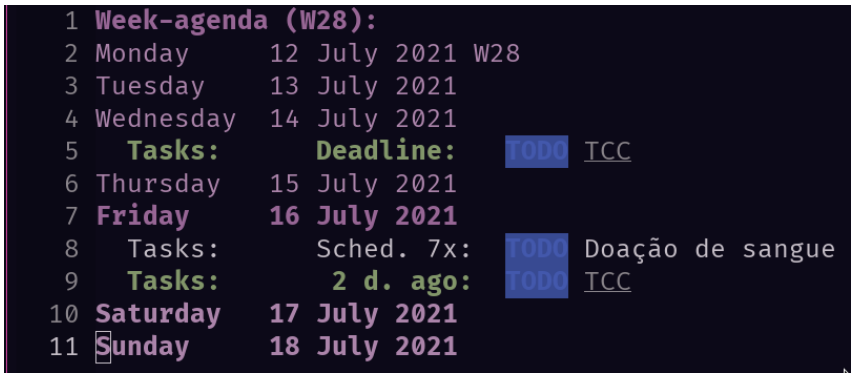
The *org-mode* organizational toolkit has been developed in such a way that reproducible research is the default (STANISIC; LEGRAND; DANJEAN, 2015)[10].

Org-mode enables *Literate Programming*[11], which is mixing code and notes - not

[10] On a similar note, watch the video entitled "Reproducible Research with GNU Emacs and Org-mode by Thibault Lestang (University of Oxford)"

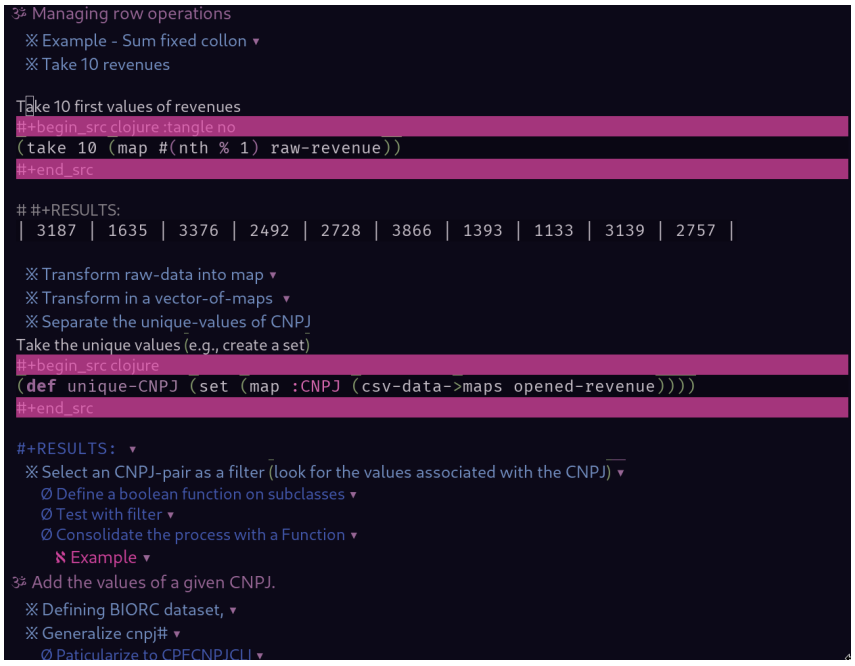
[11] conceptualized by Donald Knuth(KNUTH, 1984), and has been commonly adopted by design or naturally arose in GNU/Emacs, Org-mode and LaTeX.

Figure 23 – Org-agenda, an Orgmode feature



Resource: The authors.

Figure 24 – Literate programming in Org-mode



Mixing text, code and evaluation. Resource: The authors.

meant to be read by an interpreter/compiler. It's even possible to evaluate code inside Orgfiles. So, by reconstructing my earlier works in a orderly manner, this document was possible.

Converting .org files to .tex and thus .pdf files is as simple as calling a function in Emacs [12]. Emacs itself has been called a self-documenting system[13]. And, truly, if one wishes, all documentation of Emacs is contained inside the help-system in pure-text format. To learn a certain thing about Emacs, one would never have to leave it's environment.

Finally, translating the main .tex output into ABNT norms consisted in utilizing the

[12] Calling a function in Emacs takes one to five key-strokes depending on how you call it.
[13] https://www.emacswiki.org/emacs/SelfDocumentation

canonical features of the “abntex2” class document.

Figure 25 – ABNT specification class in LaTeX

```
\documentclass[
% -- opções da classe memoir --
12pt,                          % tamanho da fonte
openright,                     % capítulos começam em pág ímpar (insere página
vazia caso preciso)
oneside,                       % para impressão em recto e verso. Oposto a ones
ide
a4paper,                       % tamanho do papel.
% -- opções da classe abntex2 --
% chapter=TITLE,               % títulos de capítulos convertidos em letras mai
úsculas
% section=TITLE,               % títulos de seções convertidos em letras maiúsc
ulas
% subsection=TITLE,     % títulos de subseções convertidos em letras maiúsculas
% subsubsection=TITLE,% títulos de subsubseções convertidos em letras maiúsculas
% -- opções do pacote babel --
% french,                                % idioma adicional para hifenização
% spanish,                      % idioma adicional para hifenização
brazil,                        % o último idioma é o principal do documento
english,                       % idioma adicional para hifenização
]{abntex2}
```

Source: the authors.

The cover and typographical details were constrained to the particular style prescribed by the Engineering School of Lorena - University of São Paulo (EEL-USP).

4.5 Bibliographical Research

The bibliographical research was done majorly through sites, when dealing with historical data on programming or programs itself. Because these kinds of data are self-documented by the open community.

In turn, when speaking about theoretical ideas or benchmarks, the research was done by DuckDuckGo, ACM, Google Scholar or ArXiv[14]. The format found were mostly articles of books in these researches.

Vast literature was found on Julia’s DifferentialEquations.jl. Due to the fact that the Julia community arose from Academia (MIT), and it’s main goal in applied research.

Finally, some documentation was found only through GitHub, as OR-Tools and Clojure’s Clojure2D library. These were entirely electronic publications.

[14] Currently(2021) contains 1.9mi. papers openly available

5 Conclusion

Among the open source initiatives, there exist virtually an application for any one given field one may pick. That is, current research is mostly done using some kind of software application. Even more, Art, Organization, and Humanities also benefit from computation, not only Science, Technology, Engineering and Math (STEM) areas.

These open software almost majorly have State of the Art research on Computation backing them. Although the common belief may be that close software development has the upper-hand on quality and support, that is not the case.

One of the motives may be giving to the copy-left licenses coercing users of pieces of it's application to publish their software under the same license. Or, one may argue that once a person is involved in the Open Community and benefits from free and quality support from other user-developers or project maintainers. Then, nothing is more fair than contributing, when he adds to the pile of data and knowledge which boosted his way forward. Thus, the community achieves grow and continuity.

We have demonstrated many different uses of open software, with no closed sourced, paid, software. These were both in the sector of Industry and Academia applications. OR-Tools, FreqTrade, DifferentialEquations.jl and Clojure2D library developed majorly by the community use State of the Art mathematical and computing tools. OR-Tools has been, for an example, the champion in three categories of MiniZinc Challenge in 2020[1].

We also pointed some entry doors to the use of FOSS-based systems. We discussed why to use the Kernel Linux, and the GNU applications which permeates GNU/Linux system's functionalities. Furthermore, we demonstrated the benefits of inserting oneself in the social community around these open sourced initiatives et al. Finally, the effectiveness and plausibility of an running under a entirely FOSS-based system - GNU/Linux with EXWM - has used to produce this work.

[1] https://www.minizinc.org/challenge2020/results2020.html

Bibliography

ABELSON, H.; SUSSMAN, G. J. *Structure and interpretation of computer programs*. [S.l.]: The MIT Press, 1996. Citado 2 vezes nas páginas 15 and 31.

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 30.

ABRAMS, R. Optimization models for regional air pollution control. In: *Mathematical Analysis of Decision Problems in Ecology*. [S.l.]: Springer, 1975. p. 116–137. Citado na página 26.

AMIN, N.; ROMPF, T. Collapsing towers of interpreters. *Proc. ACM Program. Lang.*, Association for Computing Machinery, New York, NY, USA, v. 2, n. POPL, dez. 2017. Disponível em: <https://doi.org/10.1145/3158140>. Citado na página 16.

ARAUJO, L. C. A classe abntex2. *Documentos técnicos e científicos brasileiros*, 2018. Citado na página 30.

AVELEDA, A. A. et al. Performance comparison of scientific applications on linux and windows hpc server clusters. *Mecánica Computacional*, v. 29, n. 30, p. 2985–2997, 2010. Citado na página 22.

BERTSEKAS, D. P. *Constrained optimization and Lagrange multiplier methods*. [S.l.]: Academic press, 2014. Citado na página 25.

BIAN, Y. et al. A sectoral eco-efficiency analysis on urban-industrial symbiosis. *Sustainability*, Multidisciplinary Digital Publishing Institute, v. 12, n. 9, p. 3650, 2020. Citado na página 27.

BOURBAKI, N. Theory of sets. In: *Theory of Sets*. [S.l.]: Springer, 2004. p. 65–129. Citado 2 vezes nas páginas 12 and 14.

CHEN, C.-M.; DELMAS, M. Measuring corporate social performance: An efficiency perspective. *Production and operations management*, Wiley Online Library, v. 20, n. 6, p. 789–804, 2011. Citado na página 27.

COOK, S. The p versus np problem. *The millennium prize problems*, p. 87–104, 2006. Citado na página 25.

COSTA, M. P. et al. A socio-eco-efficiency analysis of integrated and non-integrated crop-livestock-forestry systems in the brazilian cerrado based on lca. *Journal of Cleaner Production*, Elsevier, v. 171, p. 1460–1471, 2018. Citado na página 27.

DALEY, D. J.; KENDALL, D. G. Epidemics and rumours. *Nature*, Nature Publishing Group, v. 204, n. 4963, p. 1118–1118, 1964. Citado na página 46.

DANTZIG, G. B. Reminiscences about the origins of linear programming. In: *Mathematical Programming The State of the Art*. [S.l.]: Springer, 1983. p. 78–86. Citado na página 25.

D'ELIA, M.; PACIELLO, V. Performance evaluation of labview on linux ubuntu and window xp operating systems. In: IEEE. *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*. [S.l.], 2011. p. 1494–1498. Citado na página 22.

DIRAC, P. A. The lagrangian in quantum mechanics. In: *Feynman's Thesis—A New Approach To Quantum Theory*. [S.l.]: World Scientific, 2005. p. 111–119. Citado na página 13.

DYCKHOFF, H.; ALLEN, K. Measuring ecological efficiency with data envelopment analysis (dea). *European Journal of Operational Research*, Elsevier, v. 132, n. 2, p. 312–325, 2001. Citado na página 26.

EMACS History. 2021. <https://www.emacswiki.org/emacs/EmacsHistory>. Citado na página 20.

EZICI, B.; EĞILMEZ, G.; GEDIK, R. Assessing the eco-efficiency of us manufacturing industries with a focus on renewable vs. non-renewable energy use: An integrated time series mrio and dea approach. *Journal of Cleaner Production*, Elsevier, v. 253, p. 119630, 2020. Citado na página 27.

FANG, F. et al. Cryptocurrency trading: a comprehensive survey. *arXiv preprint arXiv:2003.11352*, 2020. Citado na página 23.

FINK, M. *The business and economics of Linux and open source*. [S.l.]: Prentice Hall Professional, 2003. Citado na página 15.

FITZGERALD, B. The transformation of open source software. *MIS quarterly*, JSTOR, p. 587–598, 2006. Citado 2 vezes nas páginas 22 and 23.

GALLEGO, M. D. et al. Open source software: The effects of training on acceptance. *Computers in Human Behavior*, Elsevier, v. 49, p. 390–399, 2015. Citado 3 vezes nas páginas 17, 18, and 23.

GOWDA, S. et al. Sparsity programming: Automated sparsity-aware optimizations in differentiable programming. 2019. Citado na página 28.

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 31.

HARALAMBOUS, Y. *Fonts & encodings*. [S.l.]: " O'Reilly Media, Inc.", 2007. Citado na página 30.

HAUGE, Ø.; SØRENSEN, C.-F.; CONRADI, R. Adoption of open source in the software industry. In: SPRINGER. *IFIP International Conference on Open Source Systems*. [S.l.], 2008. p. 211–221. Citado 2 vezes nas páginas 18 and 23.

HEINZ, J. The computational nature of phonological generalizations. In: *Phonological typology*. [S.l.]: De Gruyter Mouton, 2018. p. 126–195. Citado na página 27.

HERTEL, G.; NIEDNER, S.; HERRMANN, S. Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. *Research policy*, Elsevier, v. 32, n. 7, p. 1159–1177, 2003. Citado na página 15.

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 15.

JACOBS, B. W.; KRAUDE, R.; NARAYANAN, S. Operational productivity, corporate social performance, financial performance, and risk in manufacturing firms. *Production and Operations Management*, Wiley Online Library, v. 25, n. 12, p. 2065–2085, 2016. Citado na página 27.

KEYSSER, L. T.; LENZEN, M. 1.5 c degrowth scenarios suggest the need for new mitigation pathways. *Nature communications*, Nature Publishing Group, v. 12, n. 1, p. 1–16, 2021. Citado na página 26.

KNUTH, D. E. The art of computer programming, vol 1: Fundamental. *algorithms*, p. 187, 1968. Citado na página 12.

KNUTH, D. E. Literate programming. *The computer journal*, Oxford University Press, v. 27, n. 2, p. 97–111, 1984. Citado na página 51.

KNUTH, D. E. *TEX: the Program*. [S.l.]: Addison-Wesley, 1986. Citado na página 30.

KSHETRI, N. Economics of linux adoption in developing countries. *IEEE software*, IEEE, v. 21, n. 1, p. 74–81, 2004. Citado na página 23.

LAMPORT, L. *LATEX: a document preparation system: user's guide and reference manual*. [S.l.]: Addison-wesley, 1994. Citado na página 30.

LI, Y.; TAN, C.-H.; YANG, X. It is all about what we have: A discriminant analysis of organizations' decision to adopt open source software. *Decision support systems*, Elsevier, v. 56, p. 56–62, 2013. Citado 3 vezes nas páginas 17, 18, and 23.

LINUX, list of distributions. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/List_of_Linux_distributions>. Citado 2 vezes nas páginas 19 and 20.

LUPTACIK, M. et al. *Mathematical optimization and economic analysis*. [S.l.]: Springer, 2010. Citado na página 26.

MA, Y. et al. *ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling*. 2021. Citado na página 28.

MACOS version history. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/MacOS_version_history>. Citado na página 19.

MICKWITZ, P. et al. Regional eco-efficiency indicators–a participatory approach. *Journal of Cleaner Production*, Elsevier, v. 14, n. 18, p. 1603–1611, 2006. Citado na página 27.

MICROSOFT, list of operating systems. [S.l.]: Wikimedia Foundation, 2021. <https://en.wikipedia.org/wiki/List_of_Microsoft_operating_systems>. Citado na página 19.

MONACHE, M. L. D. et al. Feedback control algorithms for the dissipation of traffic waves with autonomous vehicles. In: *Computational Intelligence and Optimization Methods for Control Engineering*. [S.l.]: Springer, 2019. p. 275–299. Citado na página 27.

MOODY, G. *Rebel code: Linux and the open source revolution*. [S.l.]: Hachette UK, 2009. Citado na página 15.

PANZA, M. The origins of analytic mechanics in the 18th century. *A history of analysis, ed. HN Jahnke, AMS*, p. 137–153, 2003. Citado na página 13.

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 15.

PINTER, C. C. *A book of set theory*. [S.l.]: Courier Corporation, 2014. Citado na página 13.

PIQUEIRA, J. R.; ZILBOVICIUS, M.; BATISTELA, C. M. Daley–kendal models in fake-news scenario. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 548, p. 123406, 2020. Citado na página 46.

PIQUEIRA, J. R. C. Rumor propagation model: an equilibrium study. *Mathematical Problems in Engineering*, Hindawi, v. 2010, 2010. Citado na página 46.

POTTS, C. et al. Harmonic grammar with linear programming: from linear systems to linguistic typology. *Phonology*, Cambridge University Press, v. 27, n. 1, p. 77–117, 2010. Citado na página 27.

RACERO, F. J.; BUENO, S.; GALLEGO, M. D. Predicting students' behavioral intention to use open source software: A combined view of the technology acceptance model and self-determination theory. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 8, p. 2711, 2020. Citado 3 vezes nas páginas 17, 18, and 23.

RACERO, F. J.; BUENO, S.; GALLEGO, M. D. Can the oss-focused education impact on oss implementations in companies? a motivational answer through a delphi-based consensus study. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 10, n. 3, p. 277, 2021. Citado 2 vezes nas páginas 17 and 23.

RACKAUCKAS, C. et al. Diffeqflux.jl-a julia library for neural differential equations. *arXiv preprint arXiv:1902.02376*, 2019. Citado na página 28.

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 28.

RACKAUCKAS, C. et al. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020. Citado na página 28.

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 28.

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 28.

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 28.

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 28.

RISTOV, S.; GUSEV, M. Performance vs cost for windows and linux platforms in windows azure cloud. In: *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*. [S.l.: s.n.], 2013. p. 214–218. Citado na página 22.

SCHIEWE, P. et al. *Integrated optimization in public transport planning*. [S.l.]: Springer, 2020. Citado na página 27.

SCHMIDT, C. Agile software development. In: *Agile Software Development Teams*. [S.l.]: Springer, 2016. p. 7–35. Citado na página 23.

SCHRAPE, J.-F. Open-source projects as incubators of innovation: From niche phenomenon to integral part of the industry. *Convergence*, SAGE Publications Sage UK: London, England, v. 25, n. 3, p. 409–427, 2019. Citado 2 vezes nas páginas 18 and 23.

S.DEVAN, S. Windows 8 v/s linux ubuntu 12.10 - comparison of the network performance. *International Journal of Research in Engineering and Technology*, v. 02, p. 577–580, 2013. Citado na página 22.

SIMONIS, H. Sudoku as a constraint problem. In: CITESEER. *CP Workshop on modeling and reformulating Constraint Satisfaction Problems*. [S.l.], 2005. v. 12, p. 13–27. Citado na página 25.

SMITH, C.; RITCHIE, S. *SICMUtils: Functional Computer Algebra in Clojure*. 2016. <http://github.com/sicmutils/sicmutils>. Citado na página 31.

SPINELLIS, D.; GIANNIKAS, V. Organizational adoption of open source software. *Journal of Systems and Software*, Elsevier, v. 85, n. 3, p. 666–682, 2012. Citado 2 vezes nas páginas 17 and 23.

STALLMAN, R. *Linux and GNU - GNU Project - Free Software Foundation*. 1997. <https://www.gnu.org/gnu/linux-and-gnu.html>. Citado na página 20.

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 20.

STANISIC, L.; LEGRAND, A.; DANJEAN, V. An effective git and org-mode based workflow for reproducible research. *ACM SIGOPS Operating Systems Review*, ACM New York, NY, USA, v. 49, n. 1, p. 61–70, 2015. Citado na página 51.

SULAIMAN, N. S.; RAFFI, A. S. H. A. Comparison of operating system performance between windows 10 and linux mint. *International Journal of Synergy in Engineering and Technology*, v. 2, n. 1, p. 92–102, 2021. Citado 2 vezes nas páginas 17 and 22.

SUSSMAN, G. J.; WISDOM, J. *Structure and interpretation of classical mechanics*. [S.l.]: The MIT Press, 2015. Citado na página 31.

SYKORA, H. T. et al. Stochasticdelaydiffeq. jl-an integrator interface for stochastic delay differential equations in julia. 2020. Citado na página 28.

TANENBAUM, A. S.; BOS, H. *Modern operating systems*. [S.l.]: Pearson, 2015. Citado na página 17.

THUBAASINI, P.; RUSNIDA, R.; ROHANI, S. Efficient comparison between windows and linux platform applicable in a virtual architectural walkthrough application. In: *Innovations in Computing Sciences and Software Engineering*. [S.l.]: Springer, 2010. p. 337–342. Citado na página 22.

TORVALDS, L. Linux: a portable operating system. *Master's thesis, University of Helsinki*, 1997. Citado na página 20.

TU, Q. et al. Evolution in open source software: A case study. In: IEEE. *Proceedings 2000 International Conference on Software Maintenance*. [S.l.], 2000. p. 131–142. Citado na página 15.

WEST, J.; DEDRICK, J. Open source standardization: the rise of linux in the network era. *Knowledge, Technology & Policy*, Springer, v. 14, n. 2, p. 88–112, 2001. Citado na página 15.

WOLFRAM, S. Cellular automata. *Los Alamos Science*, v. 9, n. 2-21, p. 42, 1983. Citado na página 48.

ZHANG, B. et al. Eco-efficiency analysis of industrial system in china: A data envelopment analysis approach. *Ecological economics*, Elsevier, v. 68, n. 1-2, p. 306–316, 2008. Citado na página 26.

ZHANG, J. et al. Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing*, Springer, v. 30, n. 4, p. 1809–1830, 2019. Citado na página 25.