

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**PATRICK FERRO RIBEIRO**

**NETLABX - FERRAMENTA VISUAL PARA CONTROLE DE AMBIENTE  
VIRTUALIZADO DE REDE**

**GUARAPUAVA**

**2022**

**PATRICK FERRO RIBEIRO**

**NETLABX - FERRAMENTA VISUAL PARA CONTROLE DE AMBIENTE  
VIRTUALIZADO DE REDE**

**NETLABX - INTERFACE FOR VIRTUALIZED NETWORK ENVIRONMENT  
CONTROL**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Professor Dr. Hermano Pereira

**GUARAPUAVA**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**PATRICK FERRO RIBEIRO**

**NETLABX - FERRAMENTA VISUAL PARA CONTROLE DE AMBIENTE VIRTUALIZADO DE REDE**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Sistemas para Internet do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná (UTFPR).

Data da aprovação: 14/dezembro/2022

---

Prof. Hermano Pereira  
Doutor  
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

---

Prof. Luciano Ogiboski  
Doutor  
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

---

Prof. Sediane Carmem Lunardi Hernandes  
Doutora  
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

GUARAPUAVA

2022

Dedico este trabalho ao meu pai Enio do Nascimento Ribeiro, e a minha mãe Irma Ferro Ribeiro (in memoriam), por terem acreditado na minha capacidade desde sempre, pela paciência e incentivo no estudo, no trabalho e na honestidade, só tenho a agradecer por tudo.

E dedico também especialmente a minha esposa Patricia Fabiane Ferro Ribeiro, que durante a minha vida acadêmica esteve presente desde o início, compartilhando de vários momentos, amizades e dificuldades enfrentadas durante minha graduação, sendo minha grande companheira, amiga e incentivadora.

Por último e não menos importantes, dedico aos meus enteados, Vicente Fabiane e Celina Fabiane, para que o incentivo e perseverança em seus estudos também sejam recompensados em um futuro próximo, desejo lhes muita sabedoria em seus caminhos.

## **AGRADECIMENTOS**

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida.

Agradeço ao meu orientador Prof. Dr. Hermano Pereira, pela sabedoria com que me guiou nesta trajetória, mesmo nos momentos quais eu estava incerto ou com dificuldades na realização do projeto, sempre acreditou e conseguiu me restabelecer no caminho correto.

Aos meus colegas de sala e de Universidade, pelos grandes momentos vivenciados juntos.

Aos professores do curso de Tecnologia em Sistemas Para Internet (TSI), todos quais tive felizmente uma ótima convivência, e puderam repassar grandes conhecimentos acadêmicos e profissionais.

Aos servidores e colegas estagiários da Universidade que tive a oportunidade de conhecer durante a realização de estágio.

Registro também, o meu reconhecimento à toda a minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização deste projeto, ou pelo meu amadurecimento profissional e pessoal, dentro e fora da Universidade.

## RESUMO

Com o crescimento da rede mundial de computadores e a sua utilização, torna-se necessário estudos e abstrações de conceitos utilizados na área. Dessa forma, surge grande demanda por aplicações relacionadas a segurança, expansão e seu manutenção. A definição de redes de computadores se dá pelo conjunto de equipamentos interligados de maneira a trocarem informações e compartilhem recursos, como exemplo: arquivos, impressoras, modems, softwares entre outros. Com isso, para obtenção e estudo do comportamento de uma rede, se faz necessário grande quantidade de equipamentos e capacidade técnica para configuração. No ambiente acadêmico essa necessidade torna o estudo oneroso. Sendo assim, uma alternativa para obtenção de um ambiente completo de rede é a simulação, onde os equipamentos não são alocados de forma física, e sim de forma virtual dentro de uma aplicação de computador, sendo que cada equipamento ou recurso necessário é criado dentro desse software capaz de simular uma máquina real. O processo de simulação traz vários benefícios, devido seu baixo investimento e melhor compreensão. Um ambiente simulado necessita quantidade de equipamentos físicos reduzidos, e sua configuração pode ser realizada de forma automatizada. Em atividades realizadas no curso de Tecnologia em Sistemas para Internet, no campus Guarapuava da Universidade Tecnológica Federal do Paraná, constatou-se a carência de um ambiente visualmente intuitivo e de simples configuração, em que qualquer acadêmico ou professor possa realizar suas simulações e testes, sem demandar tempo excedente na criação desse ambiente. Por conta disso, o presente trabalho tem como objetivo, descrever os passos seguidos durante o desenvolvimento de uma ferramenta visual, para o controle de equipamentos virtualizados. Centralizando a configuração de computadores, servidores, *switchs* e roteadores em uma única ferramenta.

**Palavras-chave:** rede; laboratório; ambiente; simulação; virtualização.

## ABSTRACT

With the growth of the world-wide computer network and its use, it becomes necessary studies and abstractions of concepts used in the area. In this way a great demand arises for applications related to security, expansion and its maintenance. The definition of computer networks is given by the set of interconnected equipment in order to exchange information and share resources, such as files, printers, modems, software, among others. With this, in order to obtain and study the behavior of a network, a large amount of equipment and technical capacity for configuration is necessary, in the academic environment this necessity makes the study costly. Thus, an alternative to obtain a complete network environment is the simulation, where the equipment is not allocated physically, but in virtual form within a computer application, every necessary equipment or resource is created within that capable software to simulate a real machine. The simulation process brings several benefits, due to its low investment and better understanding. A simulated environment requires a reduced amount of physical equipment, and its configuration can be automated. In the activities carried out in the technology course in internet systems, at the Guarapuava campus of the Federal Technological University of Paraná, the lack of a visually intuitive and simple configuration environment was observed, where any academic or teacher can perform their simulations and tests, without lost most time in creating this environment. For this reason, the present work aims to describe the steps followed during the development of a visual tool for the control of virtualized equipment. Centralizing the configuration of computers, servers, switches and routers in a single tool.

**Keywords:** network; laboratory; environment; simulation; virtualization.

## LISTA DE FIGURAS

<b>Figura 1 – Captura de Tela XBNet . . . . .</b>	<b>16</b>
<b>Figura 2 – Captura de Tela NetKit . . . . .</b>	<b>17</b>
<b>Figura 3 – Captura de Tela Cisco Packet Tracer . . . . .</b>	<b>18</b>
<b>Figura 4 – Captura de Tela GNS3 . . . . .</b>	<b>19</b>
<b>Figura 5 – Captura de Tela Marionnet . . . . .</b>	<b>20</b>
<b>Figura 6 – Processo de Criação de Ambientes XBNet . . . . .</b>	<b>24</b>
<b>Figura 7 – Proposta de Processo Para Criação de Ambientes NetLabX . . . . .</b>	<b>26</b>
<b>Figura 8 – Protótipo Tela Principal . . . . .</b>	<b>29</b>
<b>Figura 9 – Protótipo Tela Para Configuração do Equipamento . . . . .</b>	<b>30</b>
<b>Figura 10 – Cordenadas do Canvas . . . . .</b>	<b>34</b>
<b>Figura 11 – Funções do Canvas . . . . .</b>	<b>35</b>
<b>Figura 12 – Tela Principal . . . . .</b>	<b>37</b>
<b>Figura 13 – Descrição do Equipamento . . . . .</b>	<b>38</b>
<b>Figura 14 – Tela Configurações do Equipamento . . . . .</b>	<b>39</b>
<b>Figura 15 – Script Configuração de Máquina Virtual . . . . .</b>	<b>40</b>



## LISTA DE TABELAS

<b>Tabela 1 – Comparativo De Simuladores.</b> . . . . .	<b>21</b>
<b>Tabela 2 – Histórias do Usuário.</b> . . . . .	<b>27</b>

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

DNS	Domain Name System
DSL	Digital Subscriber Line
GNU	General Public License
GUI	Graphic User Interface
JSON	JavaScript Object Notation
PON	Passive Optical Network
TSI	Tecnologia em Sistemas Para Internet
UML	User Mode Linux
WiFi	Wireless Fidelity

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Objetivos</b>	<b>13</b>
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
<b>1.2</b>	<b>Metodologia</b>	<b>13</b>
<b>1.3</b>	<b>Diferencial Tecnológico</b>	<b>14</b>
<b>2</b>	<b>SIMULADORES</b>	<b>15</b>
<b>2.1</b>	<b>Sistema Atual XBNet</b>	<b>15</b>
<b>2.2</b>	<b>NetKit</b>	<b>16</b>
<b>2.3</b>	<b>Cisco Packet Tracer</b>	<b>17</b>
<b>2.4</b>	<b>GNS3</b>	<b>18</b>
<b>2.5</b>	<b>Marionnet</b>	<b>19</b>
<b>2.6</b>	<b>Comparativo De Simuladores</b>	<b>20</b>
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>23</b>
<b>3.1</b>	<b>ANÁLISE DO SISTEMA</b>	<b>23</b>
3.1.1	Processo de Criação de Ambientes XBNet	23
3.1.2	Processo Sugerido Pelo NetLabX	25
<b>3.2</b>	<b>Projeto do Sistema</b>	<b>27</b>
3.2.1	Requisitos da Ferramenta Visual NetLabX	27
3.2.2	Telas do Sistema	27
3.2.3	Protótipo Tela Principal	28
3.2.4	Para Configuração de Equipamentos	30
3.2.5	Tela Menu Opções	31
<b>3.3</b>	<b>Ferramentas</b>	<b>31</b>
3.3.1	Python	31
3.3.2	Virtualenv	31
3.3.3	TKinter	32
3.3.4	PyInstaller	32
3.3.5	Git e GitHub	32
3.3.6	Justificativa de Utilização	32

<b>3.4</b>	<b>Widgets do Tkinter</b>	<b>33</b>
3.4.1	Widgets Básicos	33
3.4.2	Widget Canvas	33
3.4.3	Manipulação de Elementos	36
<b>3.5</b>	<b>Resultado</b>	<b>36</b>
3.5.1	Tela principal	36
3.5.2	Equipamento	37
3.5.3	Tela Configuração	39
3.5.4	Salvar Laboratórios	39
3.5.5	Abrir Laboratórios	40
3.5.6	Execução da Ferramenta	40
3.5.7	Execução de Scripts	40
3.5.8	Scripts Iniciais Do Laboratório	40
<b>4</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>41</b>
<b>4.1</b>	<b>Conclusão</b>	<b>41</b>
<b>4.2</b>	<b>Tabalhos Futuros</b>	<b>42</b>
	<b>REFERÊNCIAS</b>	<b>43</b>

## 1 INTRODUÇÃO

A área de comunicação baseada em redes de computadores passou por um grande processo de transformação para que pudesse chegar ao formato que podemos visualizar diariamente em nossas residências, estabelecimentos, instituições e qualquer local que possua cobertura do serviço, mas para isso, muitos equipamentos, *softwares*, e pesquisas precisaram ser desenvolvidas para o melhoramento das redes.

O objeto de estudo de um profissional de redes, possui uma grande abrangência, como por exemplo, levantamento de custos, facilidade de acesso, padronização, segurança, portabilidade, configuração de equipamentos, entre outras atividades. Atividades estas que nem sempre estão relacionadas a somente um equipamento computacional. Em uma rede doméstica é comum a necessidade de um Modem baseado na tecnologia Digital Subscriber Line (DSL) ou Passive Optical Network (PON), Roteador Wireless Fidelity (WiFi) e os computadores ou dispositivos móveis sendo clientes conectados a esse serviço de rede, todos necessariamente precisam estar se comunicando corretamente para que exista o básico acesso a rede *Internet*.

A configuração deve ser efetuada em todos os equipamentos utilizados para viabilizar sua comunicação, o exemplo citado se refere a uma simples rede doméstica, para redes corporativas ou provedoras de acesso, a complexidade é aumentada por conta do grande número de computadores, roteadores, *switches*, *hubs*, antenas entre outros.

No atual mercado, existe uma grande demanda e procura por equipamentos de rede, mensalmente surgem novos equipamentos e *softwares* relacionados. Porém, toda essa infraestrutura de rede possui um valor elevado. Para realização de estudos e pesquisa, a facilidade dos ambientes simulados tem obtido grande sucesso no meio acadêmico, reduzindo os custos para o aprendizado prático e ampliando a capacidade técnica (BORGES; SARAIVA; CAMARGO, 2015).

A virtualização de equipamentos, auxilia no trabalho com ambientes onde haja uma diversidade de plataformas de *software* (sistemas operacionais), sem ter um grande aumento no número de plataformas de *hardware* (máquinas físicas), Assim cada equipamento virtual pode executar a sua distinta ferramenta e suas configurações específicas, para simular a iteração entre esses equipamentos e a sua comunicação de rede (CARISSIMI, 2008?).

Através do uso da virtualização, a simulação de ambientes se torna facilitada, para que cada equipamento virtualizado possa ser acessado e configurado no ambiente. Tornando possível a visualização do uso e configuração de ferramentas da área de redes, assim como o estudo de suas camadas de comunicação.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

A proposta deste trabalho foi o desenvolvimento de uma ferramenta visual para o controle e gerenciamento de equipamentos virtualizados de rede. Desta forma, simplificando o processo da criação, configuração e visualização de laboratórios utilizados no ambiente acadêmico. A ferramenta permite o seu uso em diferentes sistemas operacionais, pois executa chamadas de código do sistema operacional, ficando a cargo do utilizador a preparação do ambiente em que a ferramenta ira trabalhar.

### 1.1.2 Objetivos específicos

São objetivos específicos desse trabalho:

- Realizar a comunicação entre ações do usuário na ferramenta visual e equipamentos virtualizados, sem a necessidade de programação pelo usuário.
- Ilustrar os equipamentos de rede utilizados, assim como suas conexões, e possibilitar a simulação dos comportamentos energéticos, como ligar e desligar.
- Salvar configurações e ilustrações realizadas no ambiente criado, para sua posterior edição ou execução em outros equipamentos.
- Definir equipamentos pré-configurados, facilitando a criação de novos projetos de laboratório.

## 1.2 Metodologia

A metodologia consistiu em cinco etapas descritas a seguir:

- **Estudo de ferramentas disponíveis:** realizado com o intuito de levantar informações básicas e funcionalidades de ferramentas utilizadas para proposito semelhante, aprimorando assim a aplicação NetLabX.
- **Levantamento de requisitos:** iniciada juntamente com a escrita do projeto, avaliando as necessidades apontados pelo professor Dr. Hermano Pereira, e outras questões julgadas como necessárias após a observação de ferramentas disponíveis. O conceito de histórias do usuário foi utilizado em que foi possível definir as ações do usuário durante a utilização do *software*.

- **Protótipo de tela:** obtendo todas as histórias do usuário, fez-se possível a definição do formato de visualização da tela, utilizando a ferramenta de diagramação draw.io.
- **Definição das tecnologias a serem utilizadas:** como a ferramenta possui uma necessidade simples de visualização de tela, optou-se pela utilização da linguagem de programação Python, por conter nativamente a biblioteca Tkinter, utilizada para a construção de interfaces gráficas.
- **Desenvolvimento e teste no sistema:** consistiu na implementação das necessidades apresentadas, para construção da aplicação, utilizando as tecnologias. Posteriormente a realização de testes com a aplicação compilada para o sistema Linux.

### 1.3 Diferencial Tecnológico

A proposta deste trabalho foi o desenvolvimento de uma aplicação *desktop*, que possibilita a criação de laboratórios virtuais de rede, utilizando uma *interface* amigável para a utilização acadêmica, para que alunos e professores possam criar, salvar e executar seus projetos.

Por se tratar de uma ferramenta de código aberto, pode ser facilmente implantada e modificada por outras instituições, e/ou usos específicos.

A implementação da nova ferramenta busca apresentar de forma visual e dinâmica, ambientes de simulação de redes, eximindo os utilizadores da necessidade de programação, atendendo as necessidades apresentadas pelo projeto XBNet, que será apresentado a seguir, o qual foi aplicado no ambiente acadêmico através do Professor Dr. Hermano Pereira, que pode ser expandido para o desenvolvimento da ferramenta visual NetLabX. Sua principal característica, define como ambiente, e não somente uma aplicação, pois todos os elementos utilizados, são introduzidos em um sistema operacional voltado para o estudo do ambiente de rede, sistema este, que não é objeto de desenvolvimento do trabalho em questão, porém, contribui com o aprimoramento do processo, e prioriza a atenção somente nas atividades referentes a área redes, eximindo configurações complexas e conhecimento avançado em programação.

A ferramenta NetLabX possui afinidade a customização de equipamentos virtualizados, possibilitando com que equipamentos possam ser configurados de forma automatizada e criem ambientes interativos, comunicando-se com o utilizador.

## 2 SIMULADORES

Neste capítulo são apresentadas tecnologias que existem e estão disponíveis para seu uso, através de licenças de *software* distintas, buscando soluções para problemas que estão relacionados ao objetivo deste projeto.

### 2.1 Sistema Atual XBNet

O atual sistema desenvolvido para realização das atividades referentes as disciplinas de redes de computadores, ministradas pelo professor Dr. Hermano Pereira na UTFPR(Universidade Tecnológica Federal do Paraná) Campus Guarapuava, é intitulado de XB-Net que possui esse nome por conta de ser uma distribuição Xubuntu modificada, com finalidade de trabalhar em conjunto com a aplicação desenvolvida pelo professor.

Para que o processo de virtualização e simulação funcione, é necessário a configuração de um código escrito em linguagem Python, o qual comunica-se com o sistema e realiza as ações necessárias para a criação e configuração das máquinas virtualizadas, baseando-se nas imagens de sistema previamente carregadas no projeto.

Através dessas máquinas virtuais é possível realizar uma *interface* por linhas de comando, ou no modo de ambiente gráfico. Por conta dessas máquinas utilizarem sistema livre é possível simular exatamente os equipamentos de uma rede, sejam eles *switches*, *hubs*, computadores, servidores, e assim efetuar a execução de serviços dentro de cada máquina normalmente, possibilitando realizar atividades de configuração reais ao de um ambiente não virtualizado, de forma totalmente prática. Porém, a gestão e configuração desses ambientes requerem conhecimentos de programação, visto que para criar um novo laboratório é necessário a implementação de código programático.

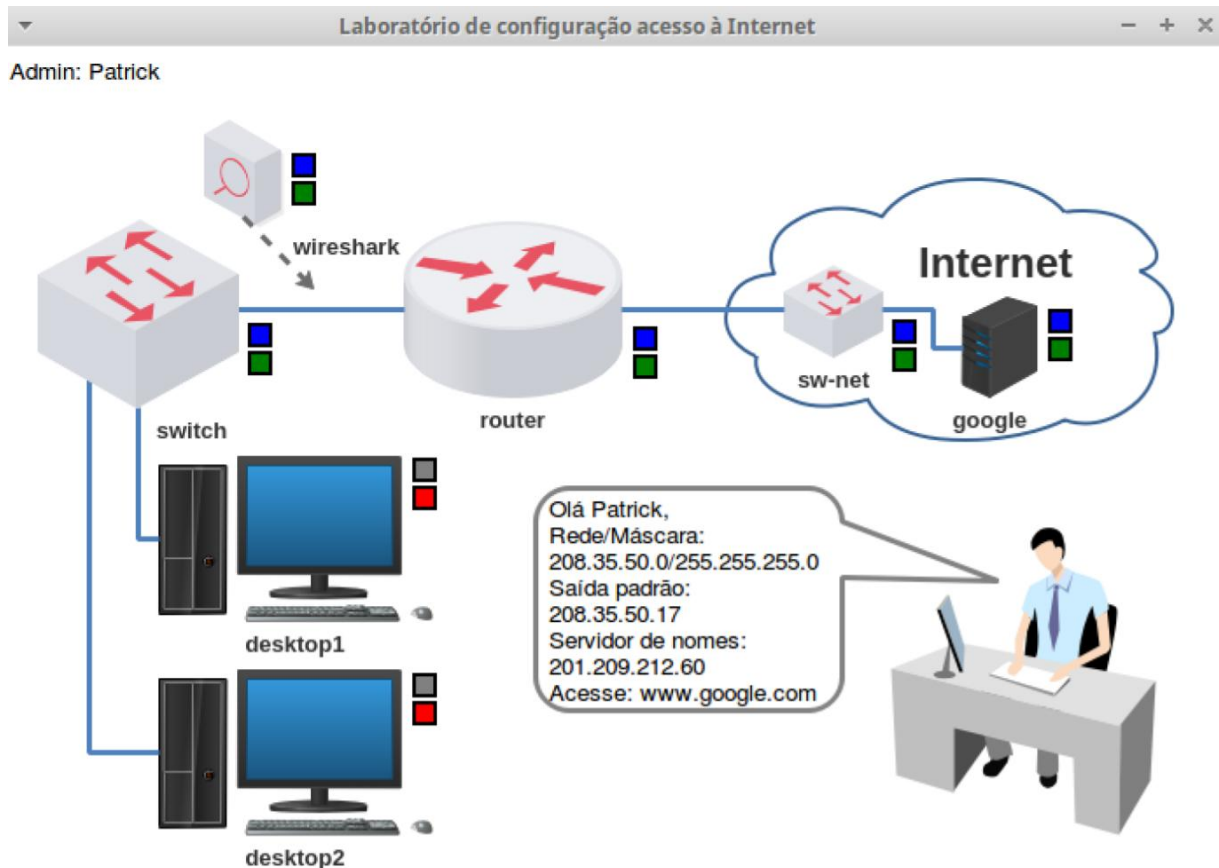
A execução dos equipamentos nesse ambiente, é realizada utilizando três ferramentas para virtualização e simulação. Primeiramente, a ferramenta VirtualBox, utilizada para criar equipamentos utilizando imagens de sistema pré-configurados, o OpenvSwitch para o comportamento de *switch* virtualizado, e User Mode Linux (UML) para utilização de diferentes distribuições Linux em forma de aplicação, isolando a aplicação do sistema operacional principal.

A Figura 1 ilustra a ferramenta XBNet, apresentando sua interface de laboratório, onde após a construção do ambiente de estudo, não é possível efetuar modificação estrutural, de forma dinâmica, sendo necessário a alteração do código de programação, para modificações estruturais e visuais.

Os equipamentos utilizados no laboratório são representados em uma imagem de fundo estática, e os botões de ação do equipamento correspondente são alocados manualmente através das coordenadas da tela.



Figura 1 – Captura de Tela XBNet



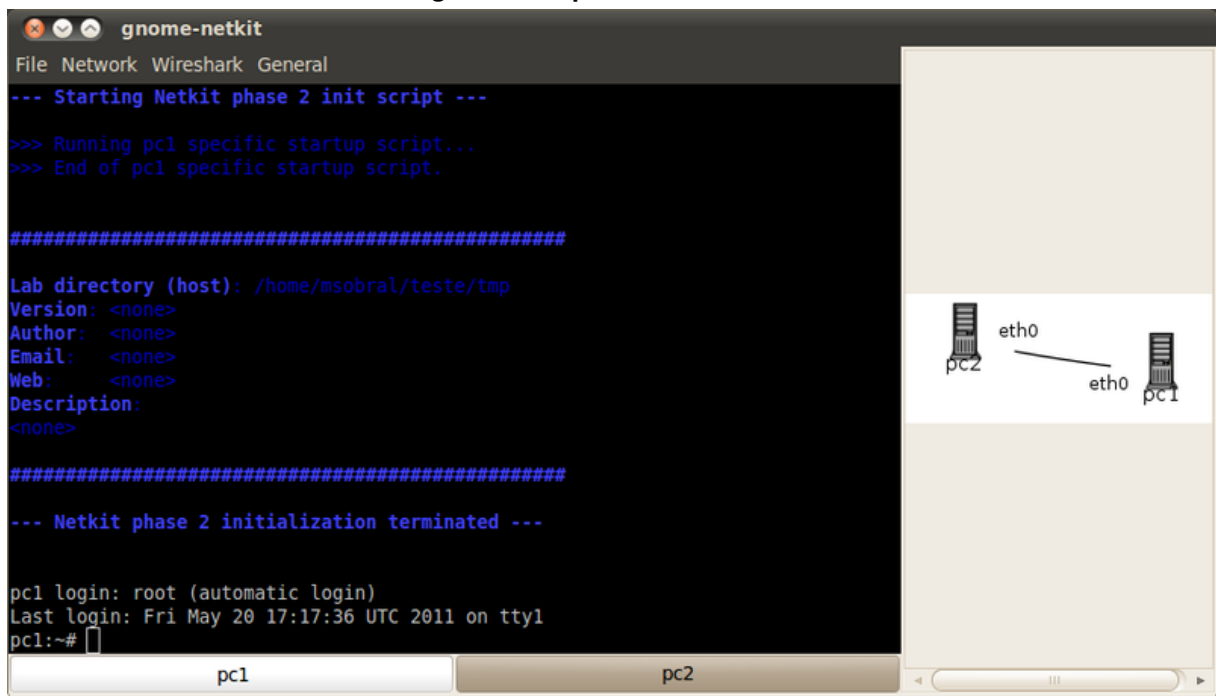
Fonte: autoria própria.

## 2.2 NetKit

A ferramenta NetKit é um *software* para realização de experimentos em redes virtualizadas que foi desenvolvido pelo IFSC (Instituto Federal de Santa Catarina) Campus São José (RIMONDINI, 2007).

Sob os termos da licença de uso GPL(General Public Licence). NetKit utiliza *software* Livre(Open Source) e tem como princípio básico a virtualização de equipamentos. Durante o uso da aplicação, o NetKit cria máquinas virtuais que possuem uma parcela dos recursos da máquina hospedeira, e por sua vez a aplicação simula a comunicação entre os equipamentos. Os equipamentos simulados utilizam uma variação do Kernel do Linux chamada UML que possui recursos básicos para a utilização dos laboratórios. A configuração desses equipamentos e a conexão entre eles é efetuada através de arquivos (BORGES; SARAIVA; CAMARGO, 2015).

Figura 2 – Captura de Tela NetKit



Fonte: (NETKIT-WIKI, 2018).

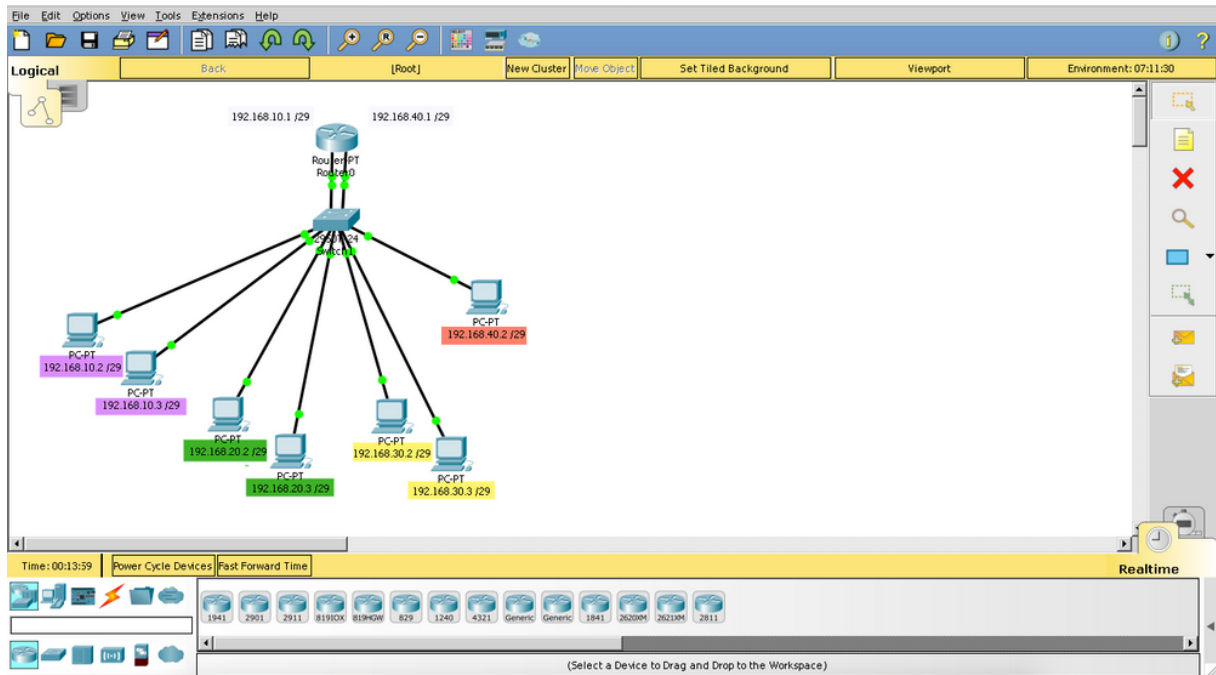
### 2.3 Cisco Packet Tracer

O *software* de simulação de ambientes de rede chamado Cisco Packet Tracer foi desenvolvido pela empresa Cisco, o qual possui grande quantidade de equipamentos relacionados a área de redes. Tem como finalidade obter o comportamento de transmissão e recebimento de pacotes dentro de um ambiente virtual, e possui uma interface que demonstra como os equipamentos são conectados fisicamente além da configuração de *software*.

É possível realizar comandos através do terminal de alguns equipamentos, mas essa ferramenta é bem limitada quanto a emulação. Seu objetivo principal é que o profissional ou estudante tenha conhecimentos nos *softwares* e equipamentos desenvolvidos pela Cisco conhecendo visualmente como os equipamentos se comunicam, e por consequência não é possível efetuar modificações nos sistemas ao contrário da ferramenta NetKit que emula um sistema Linux.

O *software* Packet Tracer é distribuído sob licença gratuita para o uso, dentro do curso introdutório em sua academia virtual (CISCO, 2010).

Figura 3 – Captura de Tela Cisco Packet Tracer



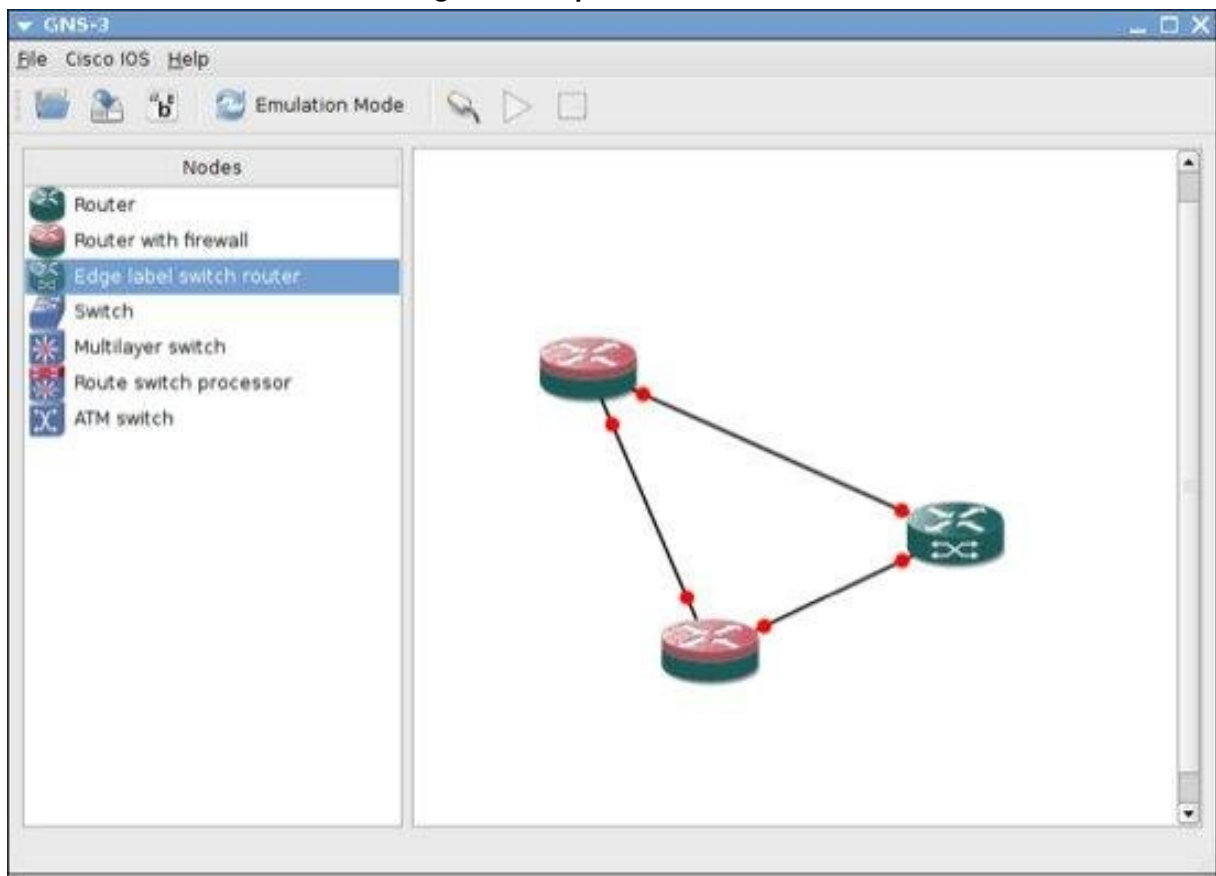
Fonte: (CISCO-COMMUNITY, 2017).

## 2.4 GNS3

É um *software* livre e possui código aberto para a comunidade. A ferramenta possui as funcionalidades de simulação e emulação. Possui suporte para ambientes de vários fornecedores.

As desvantagens se dão aos equipamentos de rede que precisam de imagens fornecidas por outros *sites*, então o usuário necessita buscar e configurar essas imagens para que os equipamentos trabalhem da forma que devem. Além disso trabalhando com emulação é necessário configurar as máquinas virtuais para seu funcionamento, alocando recursos da máquina hospedeira, tornando o processo trabalhoso (BOMBAL; DUPONCHELLE, 2019).

Figura 4 – Captura de Tela GNS3



Fonte: (GNS3-SOURCEFORGE, 2022?).

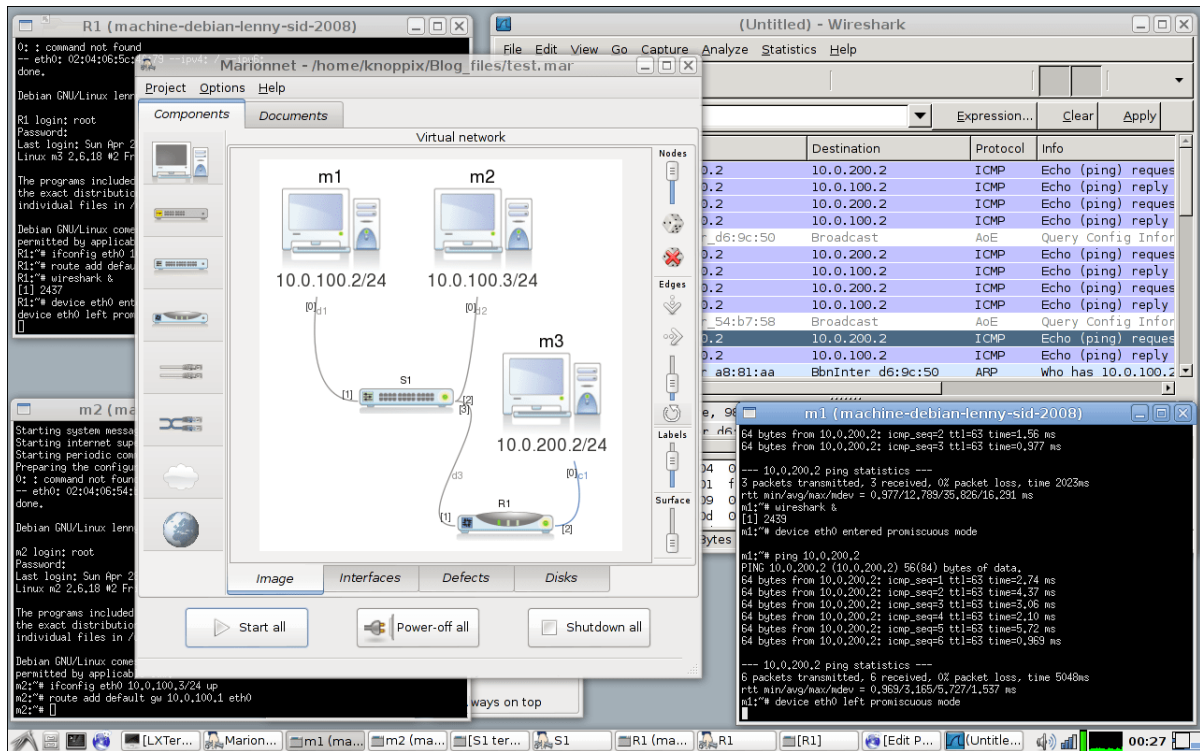
## 2.5 Marionnet

Dentre os *softwares* pesquisados o Marionnet é o qual possui maior familiaridade com o projeto desenvolvido. Esse *software* é mantido sob a licença General Public License (GNU), suas atividades principais são baseadas em simular o comportamento real de computadores e equipamentos de rede fisicamente e virtualmente.

O usuário pode adicionar novos equipamentos e controlar seus estados de energia assim como equipamentos físicos. Sua configuração pode ser feita através do terminal de cada equipamento. Que executa arquivos binários GNU/Linux não modificados baseados na arquitetura x86. Prometendo uma velocidade de emulação tão veloz quanto nativamente.

Porém, muitas de suas funcionalidades, dependem de que a máquina hospedeira possua o sistema operacional atualizado, e configurado com as dependências necessárias (LODDO; SAIU, 2008).

Figura 5 – Captura de Tela Marionnet



Fonte: (MARIONNET, 2013).

## 2.6 Comparativo De Simuladores

A Tabela 1, leva em consideração que o equipamento a ser utilizado, para execução das alternativas apresentadas, possui todos os pré requisitos instalados, como bibliotecas, arquivos, e imagens dos sistemas utilizados, atentando somente as funcionalidades e diferenciais disponíveis em cada ferramenta.

- **Ilustração do Ambiente:** Consiste na visualização do ambiente criado, em forma de equipamentos, ilustrando suas conexões, e características físicas, com finalidade de melhorar a compreensão do papel de cada equipamento dentro do ambiente virtual.
- **Equipamentos Virtualizados:** Se refere a utilização de *softwares* de virtualização, que emulam equipamentos reais, fazendo com que a configuração e simulação do ambiente, seja próxima, ou semelhante ao ambiente real.
- **Configuração Real das Interfaces de Rede:** Após a devida configuração física pela *interface* do ambiente virtual, a configuração real, leva em conta trechos de código, ou configurações no equipamento virtual, em que as interfaces de rede, são configuradas

de forma semelhante no equipamento não virtualizado.

- **Captura dos Dados Trafegados:** Aplicação que monitora e captura a comunicação entre os equipamentos utilizados, facilitando a compreensão dos protocolos e mensagens enviadas dentro da rede, através do uso da ferramenta Wireshark.
- **Interface de Criação:** Capacidade da ferramenta em que se habilita a criação de ambientes de estudo, arrastando equipamentos, e pré-configurando as suas conexões de forma visual, utilizando a *interface* da ferramenta.
- **Ambiente de Fácil Execução:** Momento em que o laboratório de estudo já está devidamente configurado pelo próprio, ou outro usuário, e somente aguarda inicialização dos equipamentos através de cliques na *interface*.
- **Ambiente de Fácil Criação:** Considera a dificuldade baixa, durante a criação de equipamentos e suas conexões, ao iniciar um novo ambiente.
- **Código Aberto:** Simuladores que possuem o código disponibilizado de forma livre, ou que possibilitam alterações e personalizações de seus arquivos.

**Tabela 1 – Comparativo De Simuladores.**

Funcionalidade	XBNet	NetKit	Cisco Packet Tracer	GNS3	Marionnet	NetLabX e XBNet
Ilustração do Ambiente	X		X	X	X	X
Equipamentos Virtualizados	X	X		X	X	X
Configuração Real das Interfaces de Rede	X				X	X
Captura dos Dados Trafegados	X				X	X
Interface de Criação			X		X	X
Ambiente de Fácil Execução	X		X		X	X
Ambiente de Fácil Criação			X		X	X
Código Aberto	X	X		X	X	X

**Fonte: autoria própria.**

O comparativo de ferramentas, levou em consideração dados que puderam ser encontrados principalmente nas páginas oficiais dos projetos, ou na inexistência da mesma em outros

artigos. Para que pudessem ser pontualmente analisados e verificados durante o desenvolvimento da ferramenta. Muitos dos simuladores apresentados necessitam de grande configuração do computador do utilizador, para que a aplicação trabalhe conforme o esperado. O objetivo da ferramenta NetLabX é apresentar o ambiente totalmente de forma visual, em conjunto com o XBNet, ou outro sistema que tenha toda a configuração básica de virtualização previamente configurado, abstraindo assim o processo de configuração do sistema operacional, no uso da ferramenta.

### 3 DESENVOLVIMENTO

Neste capítulo serão apresentadas as etapas do processo de análise e desenvolvimento, assim como a viabilidade de sua implementação.

#### 3.1 ANÁLISE DO SISTEMA

Essencialmente, o sistema deve permitir que o ambiente de aula para o estudo de redes e suas tecnologias, possa experienciar de forma prática, a realização de configuração e comunicação entre equipamentos em uma rede de computadores.

O estudo em um laboratório de rede, verifica as informações trocadas entre dois ou mais equipamentos que estão conectados, no *software* em questão o usuário deve visualizar os equipamentos da rede, e efetuar ações sobre eles, como exemplo a configuração inicial, selecionando o tipo de equipamento, o tipo de sistema que roda internamente, as aplicações e entre outros, assim como em um equipamento real, após esse processo básico, é possível inicializar esse equipamento.

O processo de inicialização, captura essas informações e envia para uma aplicação externa, que se encarrega de receber esses parâmetros e inicializar uma máquina com as características recebidas. A partir desse momento um equipamento virtual é emulado dentro do equipamento do usuário, exibindo uma interface, que pode ser visualizada via terminal ou de modo gráfico.

##### 3.1.1 Processo de Criação de Ambientes XBNet

A Figura 6, tem por finalidade a compreensão das ações e configurações necessárias para a criação de um ambiente de rede utilizando o simulador XBNet.

O primeiro passo, é a criação dos arquivos personalizados para a apresentação, e utilização do novo laboratório. Sendo uma etapa realizada exclusivamente pelo professor, por conta da necessidade de arquivos e códigos de pré-configuração.

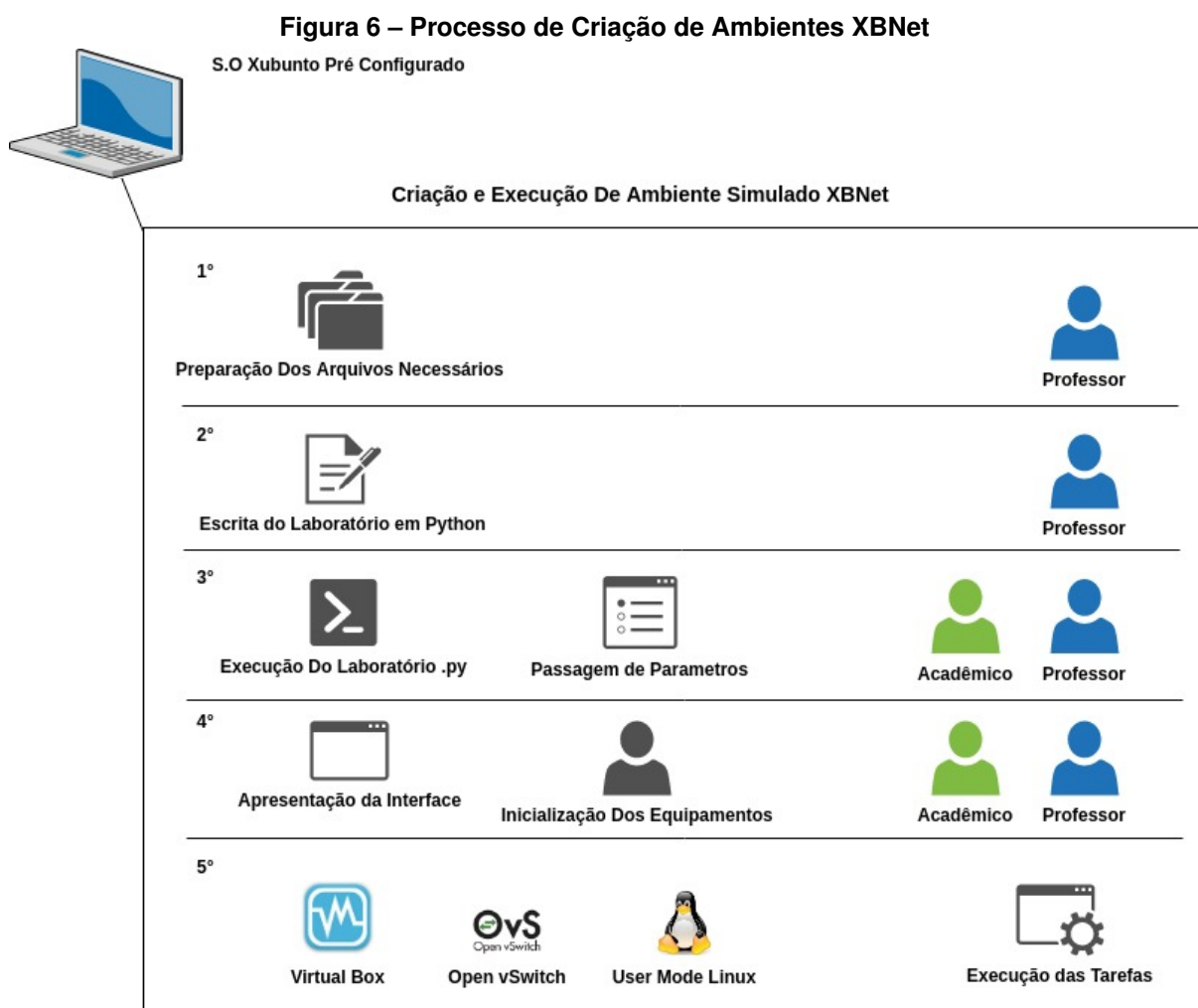
Logo a seguir, com todos os arquivos do laboratório criados, é necessário a implementação da configuração de cada equipamento a ser utilizado posteriormente, informando suas interfaces de conexão entre outras configurações. Essa etapa, realiza a configuração básica para a auto inicialização e apresentação dos equipamentos de forma simples aos acadêmicos.

Após a realização da configuração, é possível aluno e professor executar o código-fonte Python implementado, inserindo os parâmetros solicitados para criação do ambiente.

Durante o quarto processo, a *interface* do ambiente é apresentada, e aluno e professor podem selecionar os equipamentos desejados para inicialização.



O quinto e último processo, representa a ação chamada na *interface*, que de maneira silenciosa, invoca todos os processos necessários para a criação do equipamento selecionado, apresentando ao final de sua execução a tela de acesso ao sistema, no modo terminal ou interface gráfica, conforme configuração realizada no segundo processo.



Fonte: autoria própria.

### 3.1.2 Processo Sugerido Pelo NetLabX

Anteriormente apresentado o processo de criação dos ambientes, utilizando o simulador XBNet, apresentou-se as dificuldades encontradas na implementação de novos laboratórios. Na ilustração da Figura 7. Pode-se visualizar a proposta do processo realizada pelo NetLabX, levantando aspectos que facilitam as atividades desempenhadas na criação de novos ambientes.

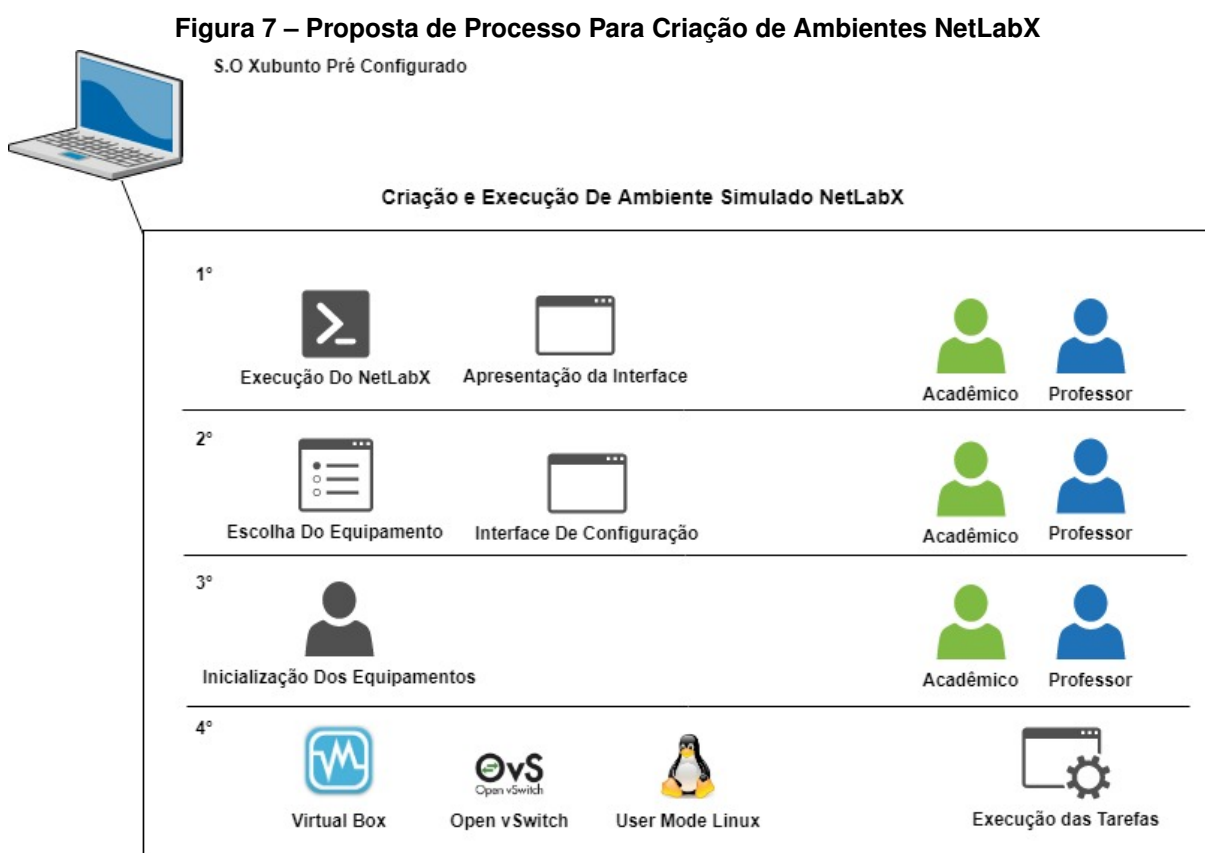
O primeiro processo, realiza a execução da aplicação, e apresentação de interface gráfica para o usuário, abstraindo toda a configuração inicial, fazendo com que este processo possa ser realizado pelos papéis de professor e aluno.

O processo seguinte, ilustra os equipamentos que podem ser criados no ambiente simulado, e realiza uma tela de pré-configuração, para relacionar equipamentos e suas conexões necessárias, simulando a forma física do ambiente.

Por seqüência, após a criação de equipamentos físicos no ambiente, habilita-se a possibilidade da sua inicialização, que efetua a execução dos códigos relacionados para cada equipamento, de forma transparente e automatizada.

Finalizando o processo de criação, após a ação de execução segue o processo normal apresentado anteriormente, em que são invocadas as aplicações de virtualização, para que os equipamentos sejam emulados e apresentados no ambiente.

Com isso, os processos que realizam as ações de configuração dos equipamentos, ficam a cargo da ferramenta, e não do utilizador, simplificando todo o processo de criação.



Fonte: autoria própria.

### 3.2 Projeto do Sistema

Nesta seção são abordados, os artefatos, desenvolvidos para a elaboração do projeto de sistema.

Serão apresentados, os requisitos da ferramenta, protótipos de tela para o sistema, assim como a descrição de suas funcionalidades e decisões arquiteturais.

#### 3.2.1 Requisitos da Ferramenta Visual NetLabX

Tendo como o objeto principal desse projeto. A interface de comunicação entre o usuário, e o código de inicialização de equipamentos, omitindo sua programação, gerando um ambiente simples e completo para o desenvolvimento de atividades e estudos.

Em conjunto com o professor, qual elaborou a iniciativa do uso de laboratórios em suas aulas, foram visualizadas, as situações em que esse ambiente poderia ajudar e acelerar o processo de ensino e aprendizagem. Para essa abstração, foram realizadas as técnicas, de elaboração das histórias do usuário, representadas no Tabela 2, onde são definidas as ações, que o aluno ou professor pode realizar durante a utilização da aplicação proposta.

**Tabela 2 – Histórias do Usuário.**

<b>História</b>	<b>Descrição</b>
H1	Como usuário, eu gostaria de criar novos projetos.
H2	Como usuário, eu gostaria de salvar os projetos dos laboratórios em forma de arquivo, para que esses arquivos possam ser modificados, e utilizados posteriormente.
H3	Como usuário, eu gostaria de criar um novo elemento, para realizar e representar ações desse equipamento no laboratório.
H4	Como usuário, eu gostaria de mover um elemento criado arrastando pela tela.
H5	Como usuário, eu gostaria de representar conexões entre os elementos, para que os equipamentos possam se comunicar na rede.
H6	Como usuário, eu gostaria de alterar as configurações de um elemento.
H7	Como usuário, eu gostaria de excluir elementos.

#### 3.2.2 Telas do Sistema

Para a elaboração do protótipo de telas, fez-se necessário a utilização da ferramenta on-line draw.io. Com essa ferramenta foi possível organizar e desenhar a *interface*, tendo como base as histórias do usuário, atendendo assim a necessidade apresentada. A produção dessas telas durante o desenvolvimento, tem como objetivo obter uma prévia, e se caso houver necessidade de remodelagem ou correção, poupar tempo de desenvolvimento.

### 3.2.3 Protótipo Tela Principal

Observando a Figura 8, se tem a representação do protótipo de tela principal da aplicação. Na parte superior da tela, existem as ações referentes a manipulação do arquivo do projeto, como criar um novo projeto, salvar e carregar projetos.

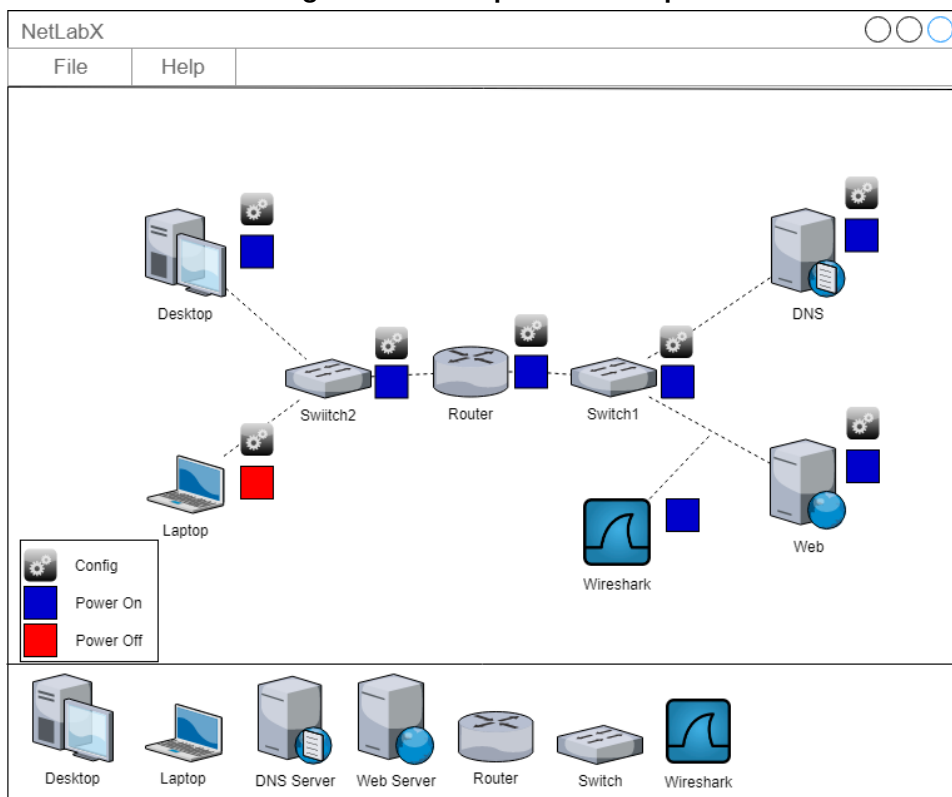
Ao centro a visualização principal para a fácil manipulação dos equipamentos, cada elemento, pode ser movido com o arraste do *mouse*. Esses equipamentos são representados por ilustrações diferentes, para gerar melhor compreensão, os elementos possuem um botão de configuração, que invoca uma nova tela para alteração dessas informações.

Os elementos que poderão ser inicializados, e fazem parte desse processo inicial do trabalho são, o computador, que possui características de um equipamento comum, para a navegação e utilização, o *notebook* que é ilustrado como laptop, assim como o computador é utilizado para navegação e utilização, o equipamento servidor Domain Name System (DNS), é o elemento para a criação de servidores de domínio, servidor *WEB* é utilizado para criação de aplicações web dentro do ambiente com o Apache, roteador é a interface de comunicação que pode interligar redes distintas, *switch* utilizado para a conexão de equipamentos dentro uma mesma rede, e por fim a ferramenta Wireshark, que não gera um sistema operacional, e sim uma aplicação, utilizada para capturar informações trocadas entre os equipamentos.

Esses equipamentos, quando executados, geram uma nova tela que é semelhante a gerada em um equipamento físico, que possui um sistema operacional Linux, podendo gerar em modo gráfico, onde o usuário utiliza teclado e *mouse* para navegar entre as telas do sistema, ou a opção de *console*, o sistema reconhece somente linhas de comando.

Na ilustração seguinte, são listados os equipamentos disponíveis para criação do ambiente, cada equipamento. Ao ser clicado, aciona a criação desse equipamento na tela, sendo visualizado na da tela, para que o aluno, ou professor possa move-lo até a posição que melhor desejar.

Figura 8 – Protótipo Tela Principal



Fonte: autoria própria.

### 3.2.4 Para Configuração de Equipamentos

A tela de configuração apresentada na Figura 9, é o momento em que o usuário estabelece como o equipamento deve ser inicializado, definindo suas informações básicas.

Equipamentos criados, somente são executados, se o usuário clicar no estado de energia do equipamento, desencadeando a ação, e gerando o código necessário para internamente executá-lo.

**Figura 9 – Protótipo Tela Para Configuração do Equipamento**

O protótipo da tela de configuração de equipamento, intitulada "Config Equipment", é exibida dentro de uma janela do NetLabX. A janela possui uma barra de menu com "File" e "Help". O formulário de configuração contém os seguintes campos e controles:

- Name:** Campo de texto com o placeholder "Name".
- Description:** Campo de texto com o placeholder "Description".
- Domain:** Campo de texto com o placeholder "domain.com".
- Parameter:** Campo de texto com o placeholder "parameter".
- Connection:** Seção com dois campos de texto contendo "ip" e "gateway", e dois menus suspensos rotulados "Interface" e "Equipment".
- Buttons:** Um botão azul "Add" ao lado dos menus de conexão e um botão azul "Save" na parte inferior direita.
- Options:** Três controles deslizantes para "Graphical Mode", "Console" e "DHCP", todos configurados para "ON".
- Category Selection:** Uma barra de seleção na base da janela com ícones e rótulos para "Desktop", "Laptop", "DNS Server", "Web Server", "Router", "Switch" e "Wireshark".

**Fonte: autoria própria.**

### 3.2.5 Tela Menu Opções

A tela de menu, possui as necessidades básicas de salvar os laboratórios do usuário, abrir projetos que já foram salvos, sair e criar um novo projeto.

## 3.3 Ferramentas

### 3.3.1 Python

A linguagem de programação Python é uma linguagem considerada de alto nível, por conta de sua simples sintaxe. Possui fácil integração em equipamentos ou sistemas operacionais distintos.

É uma linguagem interpretada, pois todos os códigos necessários para a execução de uma aplicação são passados por um interpretador, efetuando todas as chamadas necessárias para a realização da ação implementada.

Sua utilização em ambientes de rede é alta por conta do suporte que suas bibliotecas possuem, e pela simplicidade de escrita em comparação com outras linguagens. O ambiente XBNNet faz o uso desta linguagem para a inicialização dos equipamentos de forma automatizada (DOWNEY, 2012).

A utilização da linguagem Python, foi selecionada, relacionando os elementos já desenvolvidos no XBNNet, para que a integração entre as ferramentas de virtualização. Não tornem o processo demasiado complicado, tendo em vista que essa linguagem possibilita o rápido desenvolvimento dos requisitos apresentados, descartando a necessidade de mudança.

### 3.3.2 Virtualenv

Durante o desenvolvimento Python por diversos momentos existem necessidades e correções no uso de bibliotecas, onde trabalhar com o Python nativo ou instalado no próprio sistema, ocasiona certas dificuldades, pois ao executar o código-fonte em outros equipamentos, o desenvolvedor necessita pré instalar todos os seus requisitos e suas corretas versões, quais muitas das vezes geram conflitos entre si. Trabalhar desse forma dificulta o desenvolvimento, para resolver essa situação a utilização do Virtualenv aprimora a facilidade de exportação e execução de projetos.

O Virtualenv cria uma cópia das configurações em um ambiente virtual, empacota todas as necessidades básicas respeitando suas respectivas versões, recebendo seus requisitos dentro do próprio projeto (RAMOS, 2018).



### 3.3.3 TKinter

O TKinter é uma biblioteca nativa que já vem integrada a projetos que utilizam Python, aplicada ao desenvolvimento de interfaces gráficas ou Graphic User Interface (GUI), com isso é possível a criação de elementos visuais, como telas, botões e apresentações personalizadas de conteúdo, integrando o código lógico da programação, com visualizações amigáveis ao usuário. Sua simples sintaxe de escrita e integração ao código encoraja o seu uso (LABAKI, 2011?).

Com a necessidade da implementação de interfaces, o Tkinter, surge como facilitador, para a criação dos elementos visuais, que serão ilustrados no ambiente do projeto. Como a necessidade das telas apresentadas não são consideravelmente complexas, o uso da ferramenta nativa, simplifica a implementação.

### 3.3.4 PyInstaller

Durante o desenvolvimento de aplicações é normal a necessidade de configuração de ambiente para a seu correto funcionamento. Porém, uma aplicação para distribuição necessita a facilidade de execução por parte dos usuários. Com isso, o PyInstaller transforma projetos Python em arquivos executáveis.

PyInstaller lê todo o projeto verifica as necessidades para execução e empacota em um arquivo executável, qual pode ser facilmente distribuído e instalado (CORTES, 2017).

### 3.3.5 Git e GitHub

A utilização da ferramenta Git para o controle de versão do código, assim será possível realizar novas atividades e efetuar testes sem que o código em produção seja afetado, acrescentando a possibilidade de revisão e regressão caso se julgue necessário (BLISCHAK; DAVENPORT; WILSON, 2016).

O site GitHub disponibiliza de forma gratuita a submissão de repositórios para projetos de código livre, armazenando e compartilhando. Sua integração com o Git, gerencia os códigos desenvolvidos no projeto, auxiliando todo o processo.

### 3.3.6 Justificativa de Utilização

A aplicação desenvolvida, utiliza tecnologias que permitem o manutenção e atualização, facilitando a disponibilização do código de forma livre. O intuito é que usuários futuramente possam prestar experiências e até auxílio no desenvolvimento de novas funcionalidades, permitindo que mais professores e acadêmicos tenham acesso à ferramenta, assim contribuindo

com o enriquecimento do projeto, e promovendo a disseminação de conhecimento em forma prática.

A criação de arquivos de laboratório, ocorre quando a aplicação compila as informações salvas, convertendo os elementos visuais do laboratório em memória, para linhas de texto do arquivo, utilizando JavaScript Object Notation (JSON), durante a abertura desses arquivos do projeto, todas essas linhas devem ser interpretadas, para a conversão desses objetos armazenados, em instâncias da aplicação, e assim carregando todos os equipamentos configurados, possibilitando a sua visualização e execução.

### 3.4 Widgets do Tkinter

No modulo Tkinter existem diversos elementos chamados de *widgets*, termo sem tradução utilizado para referir-se a elementos da interface gráfica, quais possuem comportamentos e formas diferentes de serem apresentados em tela, com o intuito de construir a *interface* dentro da aplicação Python (LABAKI, 2011?).

#### 3.4.1 Widgets Básicos

São os elementos visualizados em tela utilizados no Tkinter, como caixas de texto, imagens, botões entre outros:

**Label:** Apresentação de texto para descrição de elementos na tela.

**Image:** Carrega um arquivo de imagem do projeto para tela da aplicação.

**Entry:** Caixa de entrada de texto para processar informações.

**Text:** Caixa de entrada de texto, expansível em maior quantidade de linhas.

**Button:** Botão para chamada de ações na aplicação.

**Frame:** Organiza elementos na tela.

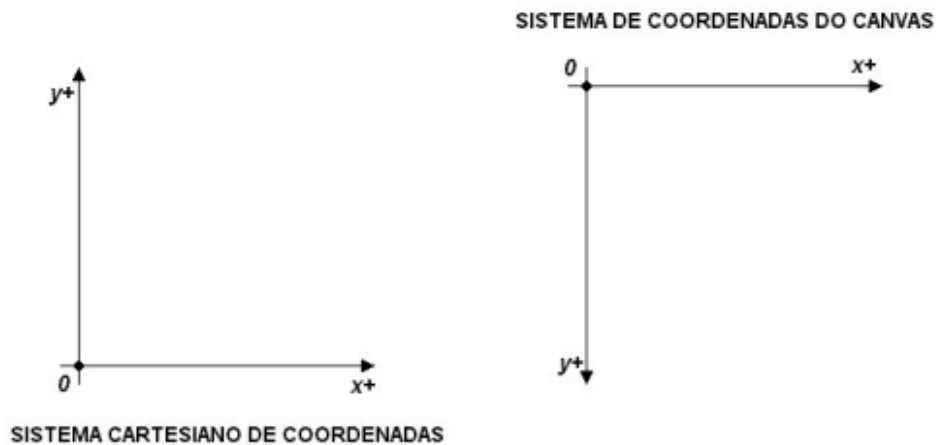
**Menu:** Cria um menu de navegação para executar ações.

#### 3.4.2 Widget Canvas

Durante a pesquisa por ferramentas e bibliotecas que auxiliariam na criação do ambiente visual, por diversas vezes o resultado apresentado foi referente a utilização do *widget* canvas. Esse *widget* possui o comportamento de possibilitar o desenho de formas geométricas, receber *widgets* em seu interior como imagens e botões, e além disso capturar eventos e ações relacionadas ao mouse e teclado do usuário.

O canvas trabalha com um sistema de coordenadas para o posicionamento de *widgets* semelhante ao plano cartesiano, porém invertido, como pode ser visualizado na figura 10 (LABAKI, 2011?).

**Figura 10 – Coordenadas do Canvas**



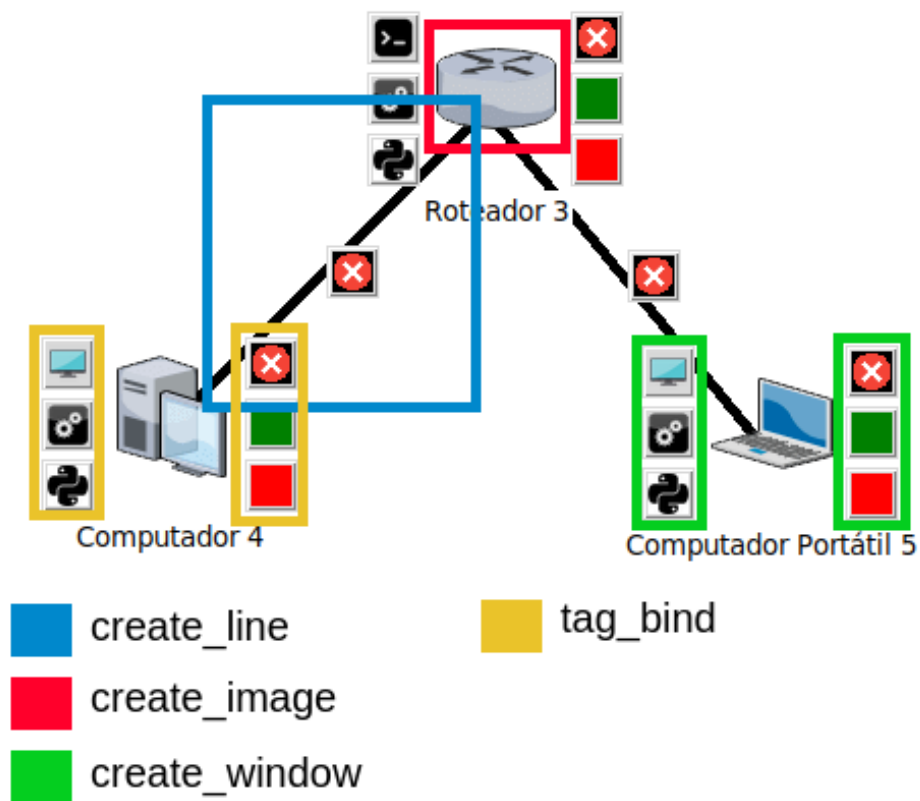
Fonte: (LABAKI, 2011?) Pág. 28.

Após instanciar o *widget* canvas no Tkinter, são possíveis algumas chamadas de função para manipular e controlar elementos em tela:

- tag\_bind:** Acrescenta eventos a determinados elementos do canvas.
- create\_window:** Adiciona elementos em determinadas coordenadas do canvas.
- create\_image:** Semelhante ao `create_window`, porém específico para imagens.
- create\_line:** Desenha uma linha em determinada coordenada.
- coords:** Recupera coordenadas de um determinado elemento do canvas.
- delete:** Remove determinado elemento do canvas.
- moveto:** Movimenta determinado elemento do canvas.

Dentro do projeto, o canvas juntamente com o conjunto de funções apresentadas, foram necessários para desenhar as linhas de conexão, adicionar as imagens de equipamentos, criar botões e adicionar funcionalidades de clique e arraste do mouse, a figura 11 ilustra as funções utilizadas em cada elemento, sinalizados por suas respectivas cores:

Figura 11 – Funções do Canvas



Fonte: autoria própria.

### 3.4.3 Manipulação de Elementos

Ao manipular elementos na *interface*, existe a necessidade de referenciar elementos para que determinadas ações modifiquem a própria *interface* durante o uso da aplicação, para isso no Tkinter existem duas formas de implementação.

O identificador *id*, onde cada elemento na tela possui seu identificador único, gerado quando elemento é adicionado a tela, através desse identificador é possível alterar esse determinado elemento, porém o uso desse identificador não se torna prático ao manipular um grupo de elementos, por conta disso a utilização de *tags* se torna fundamental no projeto, onde para um ou mais elementos pode-se adicionar uma única *tag*, e ao haver a necessidade de manipulação, esses elementos vão estar relacionados a mesma *tag* (SHIPMAN, 2013).

## 3.5 Resultado

A apresentação da ferramenta consiste no resultado obtido com a implementação do projeto NetLabX, juntamente com a visualização de suas funcionalidades.

### 3.5.1 Tela principal

Ao executar a ferramenta, o usuário pode visualizar a tela principal apresentada na figura 12, através dela todas as configurações no ambiente de simulação podem ser efetuadas.

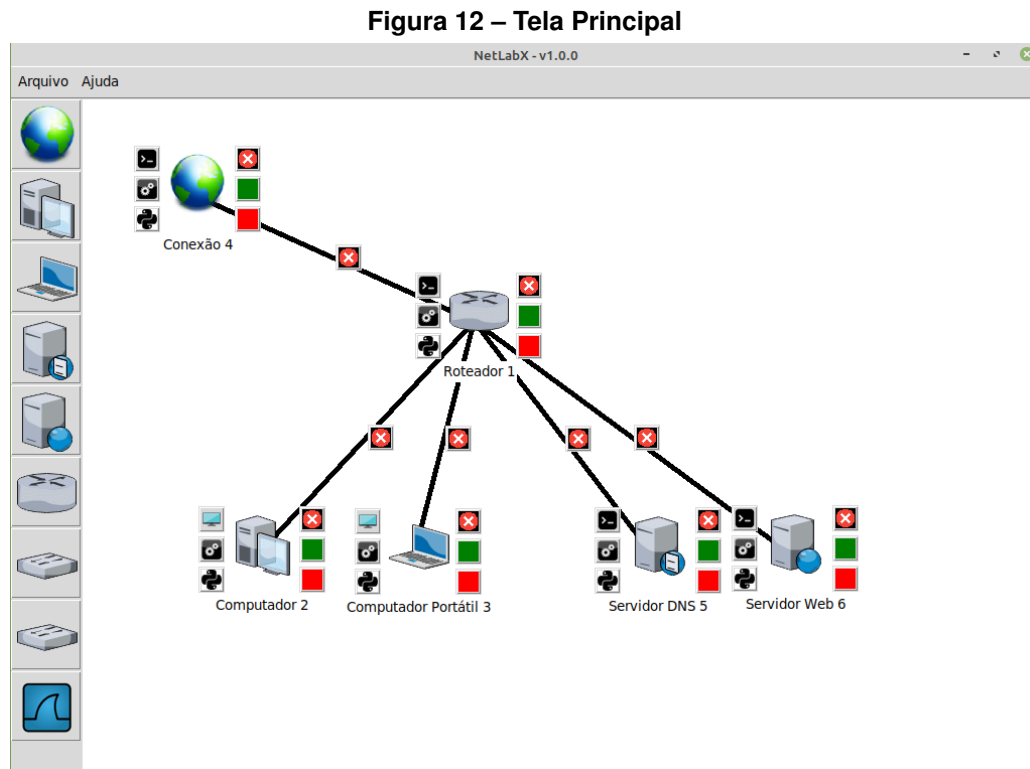
No menu superior são apresentadas as opções de arquivos e ajuda. Acessando a opção arquivos existem três funcionalidades básicas, novo para criar novos arquivos de configuração de laboratório, abrir para selecionar um arquivo previamente gerado ou repassado por outro usuário, e a opção salvar para armazenar as configurações efetuadas em tela para utilizar posteriormente. A opção ajuda aciona a tela referente a informações da aplicação

No menu lateral pode-se visualizar todos os equipamentos configuráveis do laboratório, ao clicar em uma opção de equipamento, esse novo equipamento é criado na tela da aplicação.

Na tela central, com o fundo branco, são visualizados todos os equipamentos criados pelo usuário, cada equipamento possui sua configuração específica e ações distintas configuradas em seis botões básicos, que acionam comportamentos ao serem clicados.

Ao clicar na imagem do equipamento desenhado e arrastar, é possível movimentar os elementos na tela para uma melhor organização e visualização.

Para conectar equipamentos entre si basta clicar na imagem dos equipamentos desejados, com isso a representação de linha é desenhada na tela.

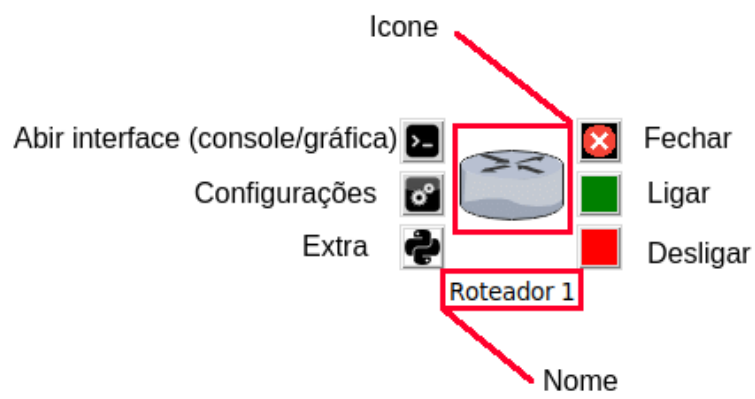


**Fonte: autoria própria.**

### 3.5.2 Equipamento

A ilustração a seguir representa o equipamento desenhado na tela do laboratório. Cada equipamento possui seis opções, um ícone e seu respectivo nome. Cada botão de opção aciona o seu respectivo *script* configurado na tela de configurações.

Figura 13 – Descrição do Equipamento



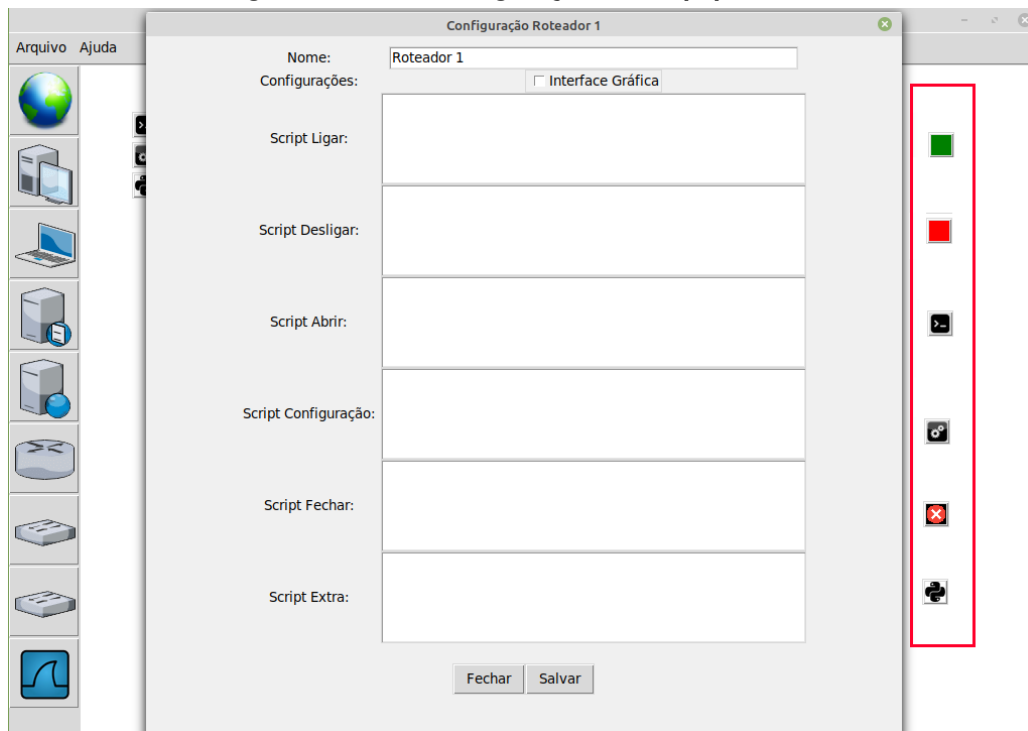
Fonte: autoria própria.

### 3.5.3 Tela Configuração

Ao acessar a opção configurações do equipamento, a tela de configurações é acionada. Nela pode-se renomear o equipamento, ativar ou desativar o ícone de interface gráfica. E configurar o código a ser utilizado em cada acionamento de botão do equipamento. A legenda destacada a direita em vermelho representa cada botão do equipamento e ao lado esquerdo o seu respectivo campo de configuração.

A configuração efetuada nesses seis campos aciona comandos a serem executados no terminal do sistema operacional, comandos esses que executam o comportamento esperado do equipamento. A quebra de linha para a execução ocorre com o carácter ";" ao final do comando, e para acionar o identificador do equipamento utiliza-se "DEVICEID".

**Figura 14 – Tela Configurações do Equipamento**



**Fonte: autoria própria.**

### 3.5.4 Salvar Laboratórios

Cada laboratório pode ser salvo e executado posteriormente, ao utilizar a opção de salvar, todos os elementos criados em tela e suas devidas coordenadas são convertidas em JSON e salvas mediante escolha de destino do arquivo.



### 3.5.5 Abrir Laboratórios

Os laboratórios salvos podem ser recarregados, ao selecionar o arquivo de configuração todas as funcionalidades e configurações são carregadas novamente e disponibilizadas em tela.

### 3.5.6 Execução da Ferramenta

Uma versão da ferramenta pode ser encontrada na página principal do GitHub do projeto, qual pode ser executada em qualquer equipamento Linux.

### 3.5.7 Execução de Scripts

A ferramenta possibilita a execução de *scripts*, onde para cada equipamento e cada ação de botão são possíveis diversas configurações. Abaixo na figura 15, um exemplo de script de inicialização de equipamento.

**Figura 15 – Script Configuração de Máquina Virtual**

```
1 vboxmanage import /opt/vbox/slitaz.ova > /dev/null 2> /dev/null;
2 vboxmanage modifyvm slitaz --name DEVICEID 2> /dev/null;
3 vboxmanage modifyvm DEVICEID --vrdeaddress 127.0.0.1 --vrde on --vrdeport 50001 2> /dev/null;
4 vboxmanage modifyvm DEVICEID --bridgeadapter1 enp0s3 2> /dev/null;
5 vboxmanage modifyvm DEVICEID --macaddress1 "0021b0c50001" 2> /dev/null;|
```

**Fonte: autoria própria.**

No *script* acima, as ações solicitam ao Vboxmanage (Ferramenta do Virtual Box), configurações básicas para que a máquina virtual seja criada pelo botão da ferramenta. Com isso a junção e utilização de configurações, podem abranger vários usos.

### 3.5.8 Scripts Iniciais Do Laboratório

Na raiz do projeto existe o arquivo *devices.conf* nesse arquivo é possível visualizar todos os equipamentos configurados inicialmente na abertura do laboratório, essa lista de equipamentos pode ser editada, assim como todos os *scripts* inicialmente configurados, fazendo com que o laboratório mude as ações de seus botões e equipamentos.

## 4 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões do presente trabalho, e as recomendações para a continuidade dos trabalhos no aprimoramento da ferramenta NetLabX.

### 4.1 Conclusão

O trabalho corrente, apresentou o projeto de desenvolvimento de um sistema *desktop*, para permitir que acadêmicos e professores, possam realizar suas atividades com maior facilidade e compreensão, visto que a área de redes, é uma grande porta de conhecimento em que os profissionais envolvidos na tecnologia da informação necessitam.

A implementação da ferramenta no ambiente da universidade, possibilita que uma maior quantidade de acadêmicos, tomem conhecimento sobre os conceitos e técnicas, tendo em vista que o Curso de Tecnologia em Sistemas para Internet somente introduz o aluno na comunicação entre computadores, deixando a maior quantidade de conteúdo para as áreas de desenvolvimento, mas a área de rede, também possui grande quantidade de demanda por profissionais de desenvolvimento, implementando soluções para facilidade de integração entre sistemas, e o trefego de informações.

Durante o estudo, é possível compreender conceitos de aplicações e serviços que são utilizados diariamente, e suas particularidades, introduzindo o acadêmico ao uso de sistemas operacionais de livre distribuição, que são utilizados tanto no ambiente de produção, como no ambiente de desenvolvimento em empresas da área.

Com relação a aplicação desenvolvida, muitos caminhos poderiam ser tomados e apresentados. Durante o desenvolvimento, uma das principais dificuldades foi a representação de exemplos. Muitas das ferramentas desenvolvidas em Tkinter não enfatizam os resultados, onde cada funcionalidade necessitava de grande pesquisa, após avançada essa dificuldade em como criar elementos em tela que pudessem ser reordenados e redesenhados, o projeto fluiu com uma maior tranquilidade, tornando fácil o entendimento e desenvolvimento.

As vantagens do uso da ferramenta estão principalmente relacionadas a como o ambiente ilustra os equipamentos utilizados em sala de aula, todos os elementos podem ser modificados e editados assim como o projeto pré definido, o ambiente inicial utilizado no XBNet possui suas ferramentas estáticas, onde o professor necessitou desenvolver cada ilustração a ser apresentada em sala de aula.

A execução de *scripts* no terminal do Linux acabou por limitar algumas funcionalidades que poderiam ser previamente implementadas, como a configuração das conexões dos equipamentos, que no projeto atual somente são visuais, porém com o arquivo de configurações de equipamentos o projeto pode ser atualizado para que alunos possam criar o seu laboratório.

Ao final pode-se concluir que a ferramenta visual cumpriu com seus objetivos principais,

e pode ser utilizada para representação ilustrada de laboratórios virtualizados, ao ser distribuída publicamente, onde por ser de código aberto pode seguir seu desenvolvimento continuando sua melhoria das atividades realizadas em sala de aula.

## 4.2 Trabalhos Futuros

Como apresentado anteriormente, durante a fase de desenvolvimento a dificuldade maior de desenho e relação dos equipamentos na ferramenta, ocasionou grande tempo de estudo para o avanço primário do projeto.

Algumas das funcionalidades que poderiam estar mais completas e funcionais seriam o relacionamento de conexões utilizando a *interface*, na ferramenta atual esse relacionamento ocorre apenas de forma visual, onde para facilidade de configuração poderia ser relacionado as *interfaces* em tela com a configuração dos *scripts*.

A geração de configurações básicas, em muitos *scripts* de laboratório ou criação de equipamentos se fazem necessários a geração de números de IP e máscaras de rede, com isso a ferramenta poderia gerar aleatoriamente esses números a serem utilizados.

A utilização de imagens e outras descrições em modo de texto, com a finalidade melhorar a visualização dos laboratórios utilizados, como exemplo ilustrar ou separar os diferentes setores de uma empresa, ou de diferentes filiais, apresentar o IP de equipamentos configurados entre outros itens.

## REFERÊNCIAS

- BLISCHAK, J.; DAVENPORT, E.; WILSON, G. **A Quick Introduction to Version Control with Git and GitHub**. 2016. Disponível em: <https://journals.plos.org/ploscompbiol/article/authors?id=10.1371/journal.pcbi.1004668>. Acesso em: 15 de abril de 2019.
- BOMBAL, D.; DUPONCHELLE, J. **Getting Started with GNS3**. 2019. Disponível em: [https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9\\_aLY8kkdhgaMB0wPCz8a38/index.html](https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.html). Acesso em: 15 de abril de 2019.
- BORGES, V.; SARAIVA, O.; CAMARGO, L. **L.S.de A. de. Software simuladores de rede: análise comparativa para apresentação de funcionalidades e benefícios**. in: Iii simtec – simpósio de tecnologia da fatec taquaritinga. 2015.
- CARISSIMI, A. **Virtualização: da teoria a solução**. 2008? Disponível em: <https://www.gta.ufrj.br/ensino/CPE758/artigos-basicos/cap4-v2.pdf>. Acesso em: 19 de Dezembro de 2022.
- CISCO. **Cisco Packet Tracer Data Sheet**. 2010. Disponível em: [https://www.cisco.com/c/dam/en\\_us/training-events/netacad/course\\_catalog/docs/Cisco\\_PacketTracer\\_DS.pdf](https://www.cisco.com/c/dam/en_us/training-events/netacad/course_catalog/docs/Cisco_PacketTracer_DS.pdf). Acesso em: 17 de abril de 2019.
- CISCO-COMMUNITY. 2017. Disponível em: <https://community.cisco.com/t5/switching/general-packet-tracer-network-problem/td-p/3297317>. Acesso em: 08 de Dezembro de 2022.
- CORTES, D. **PyInstaller Documentation: Release 3.3**. 2017. Disponível em: [https://pyinstaller.org/\\_downloads/en/v3.3/pdf/](https://pyinstaller.org/_downloads/en/v3.3/pdf/). Acesso em: 09 de Dezembro de 2022.
- DOWNEY, A. **Think Python: How to think like a computer scientist**. 2012. Disponível em: <http://greenteapress.com/thinkpython/thinkpython.pdf>. Acesso em: 23 de Junho de 2019.
- GNS3-SOURCEFORGE. 2022? Disponível em: <https://sourceforge.net/projects/gns-3/>. Acesso em: 08 de Dezembro de 2022.
- LABAKI, J. **Introdução Ao Python: Módulo c**. 2011? Disponível em: <https://www.dcc.ufrj.br/~fabiom/mab225/tutorialtkinter.pdf>. Acesso em: 23 de Junho de 2019.
- LODDO, J.-V.; SAIU, L. **Marionnet: a virtual network laboratory and simulation tool**. p. 1–3, 2008.
- MARIONNET. 2013. Disponível em: <https://www.brianlinkletter.com/2013/05/marionnet-network-simulator-test-drive/>. Acesso em: 08 de Dezembro de 2022.
- NETKIT-WIKI. 2018. Disponível em: <https://wiki.sj.ifsc.edu.br/wiki/index.php/Netkit2>. Acesso em: 08 de Dezembro de 2022.
- RAMOS, V. **PYTHON E VIRTUALENV: COMO PROGRAMAR EM AMBIENTES VIRTUAIS**. 2018. Disponível em: <https://pythonacademy.com.br/blog/python-e-virtualenv-como-programar-em-ambientes-virtuais>. Acesso em: 09 de Dezembro de 2022.
- RIMONDINI, M. **Emulation of computer networks with netkit**. p. 3–5, 2007.
- SHIPMAN, J. W. **Tkinter 8.5 reference: agui for python**. 2013. Disponível em: <https://tkdocs.com/shipman/tkinter.pdf>. Acesso em: 08 de Dezembro de 2022.