

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COINT - TECNOLOGIA EM SISTEMAS PARA INTERNET
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

PATRICK FERRO RIBEIRO

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

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2019

PATRICK FERRO RIBEIRO

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

Projeto de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso de Tecnologia em Sistemas para Internet - TSI - da Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Guarapuava, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Professor Me. Hermano Pereira

GUARAPUAVA
2019

RESUMO

RIBEIRO, Patrick. NetLabx - Ferramenta Visual Para Controle de Ambiente Virtualizado de Rede. 2019. 23 f. Projeto de Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

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 compartilhar 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, 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, onde qualquer acadêmico ou professor possa realizar suas simulações e testes, sem demandar tempo excedente na criação desse ambiente.

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

ABSTRACT

RIBEIRO, Patrick. NetLabx - Interface for Virtualized Network Environment Control. 2019. 23 f. Projeto de Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

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.

Keywords: network; laboratory; environment simulation; virtualization;

LISTA DE FIGURAS

Figura 1 – Captura de Tela XBNet	5
Figura 2 – Captura de Tela NetKit	6
Figura 3 – Captura de Tela Cisco Packet Tracer	7
Figura 4 – Captura de Tela GNS3	8
Figura 5 – Captura de Tela Marionnet	9
Figura 6 – Processo de Criação de Ambientes XBNet	15
Figura 7 – Proposta de Processo Para Criação de Ambientes NetLabX	16
Figura 8 – Protótipo da Tela Principal	19
Figura 9 – Protótipo da Tela de Configuração do Equipamento	20
Figura 10 – Diagrama de classes	21

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.1.1 OBJETIVO GERAL	2
1.1.2 OBJETIVOS ESPECÍFICOS	2
1.2 METODOLOGIA	2
1.3 DIFERENCIAL TECNOLÓGICO	3
2 – SIMULADORES	4
2.1 Sistema Atual XBNet	4
2.2 NetKit	5
2.3 Cisco Packet Tracer	6
2.4 GNS3	7
2.5 Marionnet	8
2.6 Comparativo De Simuladores	9
3 – FERRAMENTAS	11
3.1 Python	11
3.1.1 TKinter	11
3.2 Metodologia Scrum	11
3.2.1 Papéis da Equipe Scrum	12
3.2.2 Artefatos do Scrum	12
3.2.3 Eventos do Scrum	12
3.2.4 Uso do Scrum No Projeto	12
3.3 Trello	12
3.4 Git e GitHub	13
3.5 Considerações	13
4 – DESENVOLVIMENTO PRELIMINAR	14
4.1 ANÁLISE DO SISTEMA	14
4.1.1 Processo de Criação de Ambientes XBNet	14
4.1.2 Processo Sugerido Pelo NetLabX	15
4.1.3 Requisitos da Ferramenta Visual NetLabX	16
4.2 PROJETO DO SISTEMA	17
4.2.1 Decisões Arquiteturais	17
4.2.2 Protótipo das Telas do Sistema	18
4.2.3 Tela Principal	18

4.2.4	Tela Para Configuração de Equipamentos	19
4.2.5	Projeto Diagrama de classes	20
4.3	PLANEJAMENTO DO TRABALHO	22
4.4	CONSIDERAÇÕES FINAIS	22
	Referências	23

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 DSL (Digital Subscriber Line), Router WiFi (Wireless Fidelity) e os computadores ou dispositivos móveis que serão clientes desse serviço de rede, todos necessariamente precisam estar se comunicando corretamente para que exista o acesso a rede e seus serviços.

A configuração deve ser efetuada em todos os equipamentos utilizados para viabilizar sua comunicação, o exemplo citado se refere a uma 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 mercado atual existe uma grande demanda e procura por equipamentos de rede, e mensalmente surgem novos equipamentos e *softwares* relacionados.

Mas 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).

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Desenvolvimento de interface gráfica para o gerenciamento de equipamentos virtualizados de rede. Simplificando o processo da criação de laboratórios utilizados no ambiente acadêmico.

1.1.2 OBJETIVOS ESPECÍFICOS

São objetivos específicos desse trabalho:

- Realizar a comunicação entre interface e virtualização, sem a necessidade de programação pelo usuário;
- Ilustrar os equipamentos de rede utilizados, assim como suas conexões, e possibilitar a gerência dos comportamentos energéticos, como ligar e desligar.
- Salvar configurações e ilustrações do ambiente criado em arquivos, para edição ou execução posterior.
- Definir ambientes pré-configurados, facilitando a criação de novos projetos;

1.2 METODOLOGIA

A metodologia consiste em seis etapas descritas a seguir:

- **Levantamento de requisitos:** foi inicializado juntamente com a escrita do projeto, avaliando as necessidades apontados pelo professor, utilizando o conceito de histórias do usuário, onde se é possível definir as ações que o usuário deseja realizar, 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:** em breve análise, como o projeto 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 do sistema:** consiste na implementação das necessidades apresentadas, para construção da aplicação, utilizando as tecnologias definidas. Para o gerenciamento do tempo e da progressão do desenvolvimento, serão utilizados o *framework* Scrum, e a ferramenta on-line Trello, para a visualização das atividades, após o encerramento de uma atividade se faz necessário o teste, para a validação e adição ao repositório do projeto.

- **Implementação:** adição do *software* ao sistema operacional portátil, utilizado pelo professor em sala, para a disponibilização da ferramenta.
- **Análise do resultados:** análise dos resultados obtidos com o desenvolvimento e implantação.

1.3 DIFERENCIAL TECNOLÓGICO

A proposta deste trabalho, é 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 introduzido no ambiente acadêmico, e tem obtido ótimos resultados, que poderão ser expandidos com o desenvolvimento da ferramenta NetLabX.

Sua principal característica, define como ambiente, e não somente uma aplicação, pois todos os elementos utilizados, serão 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 melhoramento do processo, e prioriza a atenção somente nas atividades referentes a área redes, eximindo configurações complexas.

A ferramenta NetLabX deverá possuir 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 Me. Hermano Pereira na UTFPR(Universidade Tecnológica Federal do Paraná) Câmpus Guarapuava, é intitulado de XBNet 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.

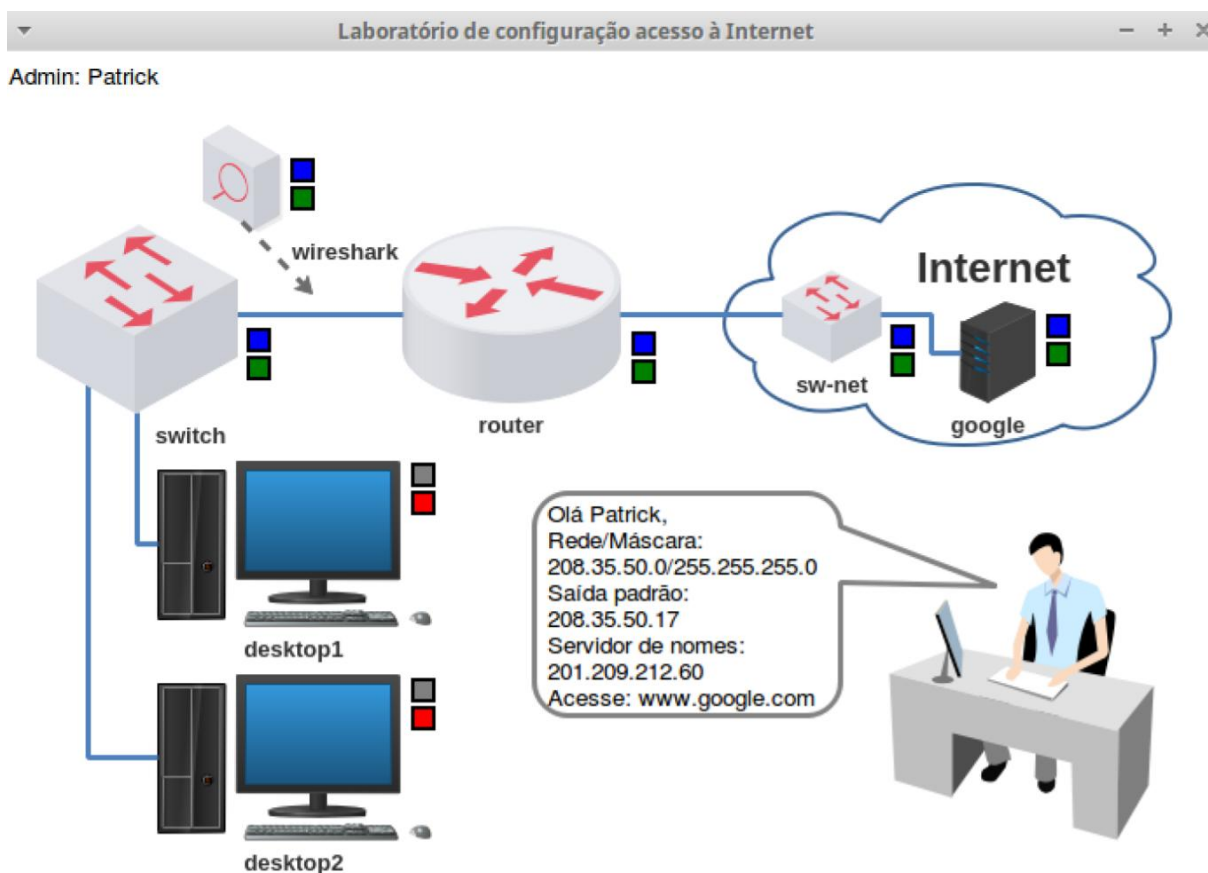
Porém a gestão e configuração desses ambientes requerem conhecimentos de programação, visto que para criar um novo laboratório é necessário a implementação de código programático.

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 para utilização de diferentes distribuições Linux em forma de aplicação, isolando a aplicação do sistema operacional principal.

A figura a seguir 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



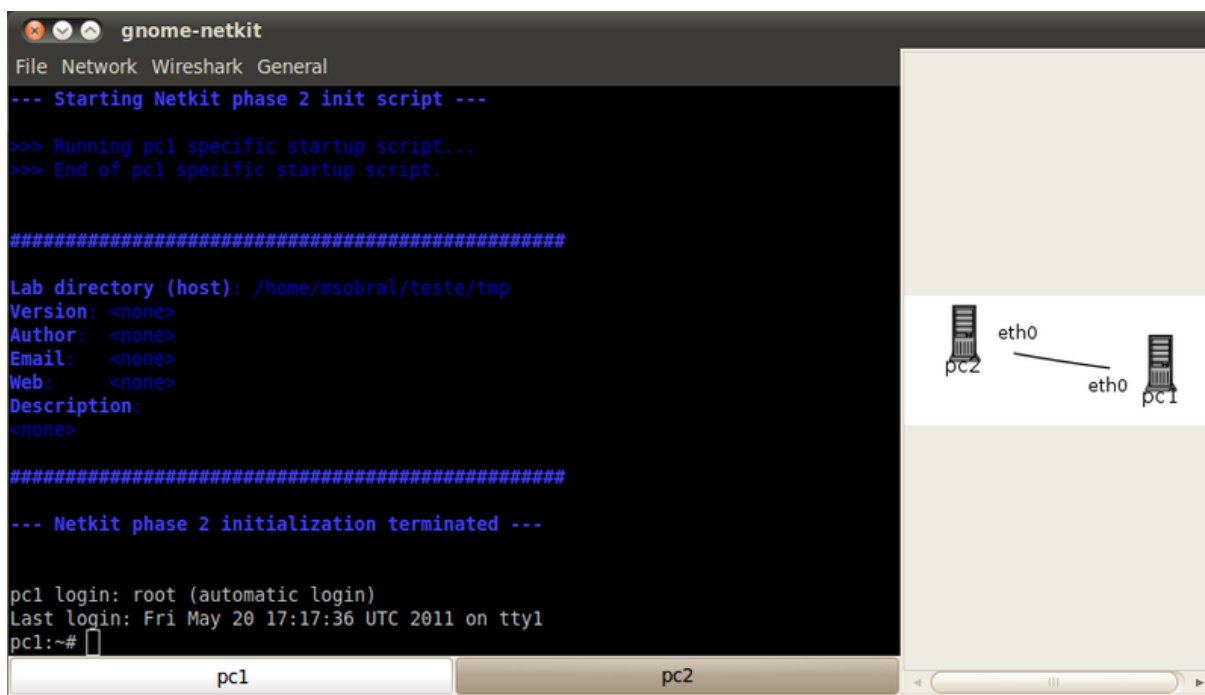
Fonte: Autor

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) e utiliza *software* Livre(Open Source), tem como princípio básico a virtualização de equipamentos, o computador que gerencia a aplicação 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 (User Mode Linux) 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: <https://wiki.sj.ifsc.edu.br/wiki/index.php/Netkit2>

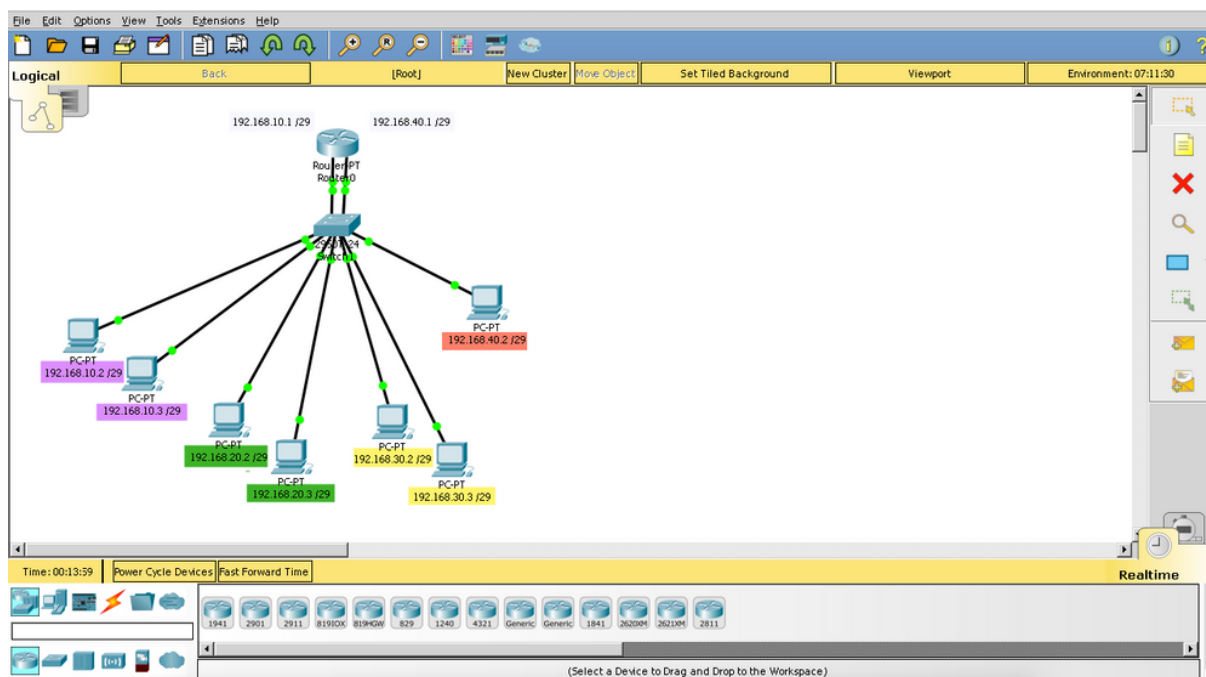
2.3 Cisco Packet Tracer

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 equipamento 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:

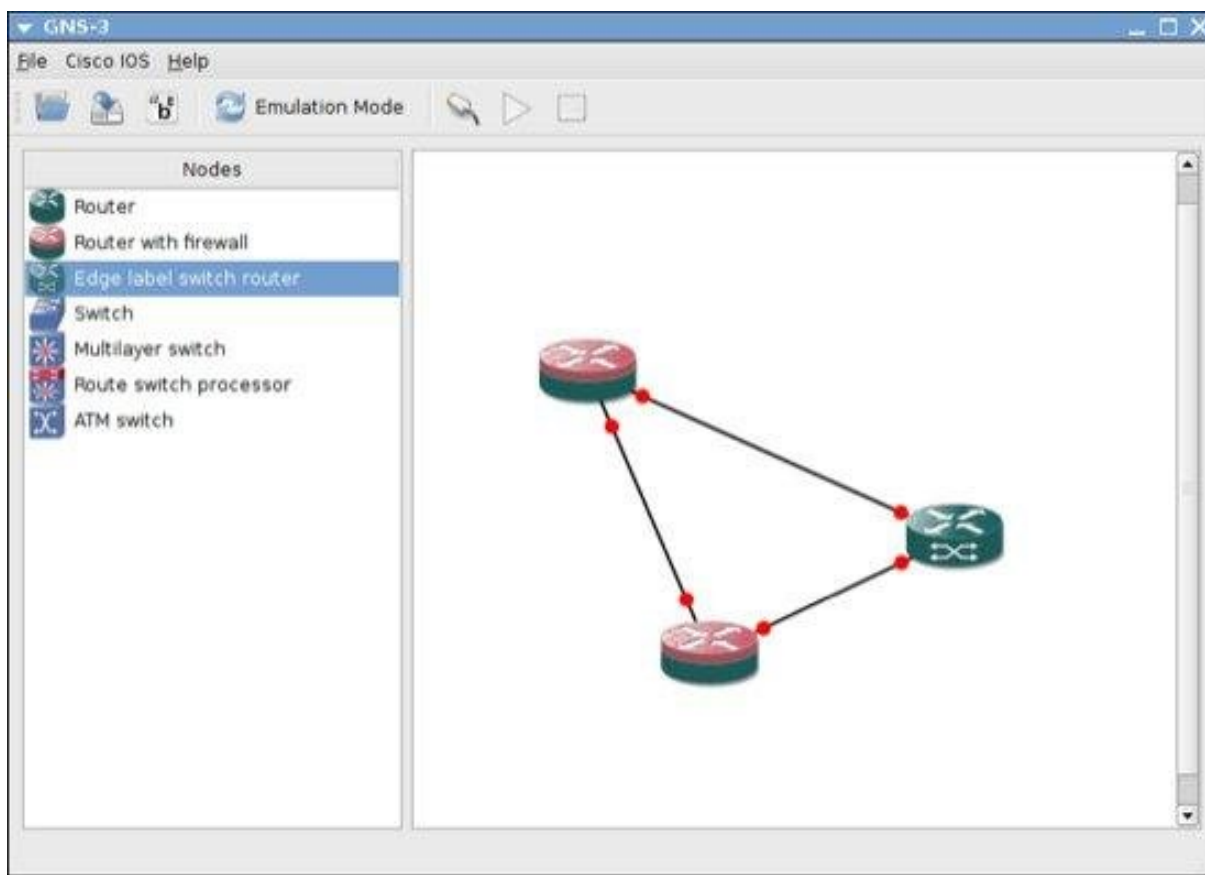
<https://community.cisco.com/t5/switching/general-packet-tracer-network-problem/td-p/3297317>

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: <https://sourceforge.net/projects/gns-3/>

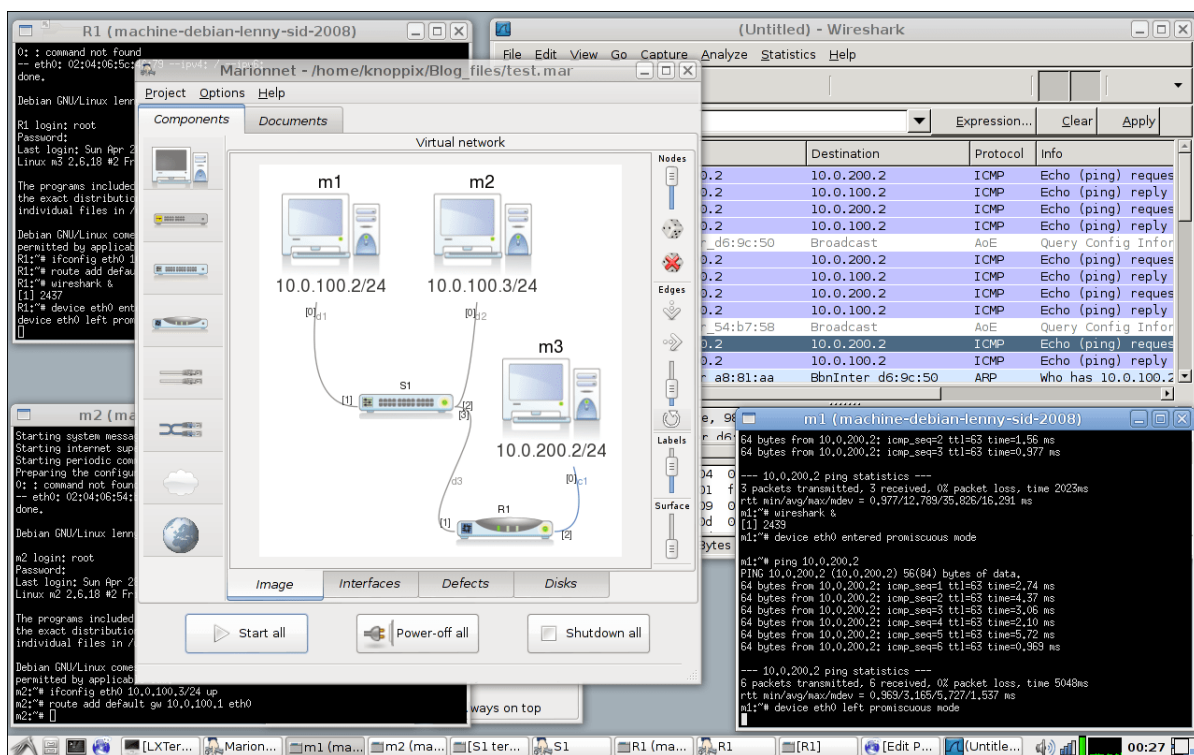
2.5 Marionnet

Dentre os *softwares* pesquisados o Marionnet é o qual possui maior familiaridade com o projeto a ser desenvolvido, esse *software* é mantido sob a licença GNU (General Public License), 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 reais, sua configuração pode ser feita através do terminal de cada equipamento, que executa arquivos binários GNU/Linux não modificados baseados na arquitetura x86, que promete 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: <https://www.brianlinkletter.com/marionnet-network-simulator-test-drive/>

2.6 Comparativo De Simuladores

O Quadro 1 de comparativos a seguir, 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, atentando somente as funcionalidades disponíveis por cada simulador.

- **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.
- **Equipamentos Virtualizados:** Se refere a utilização de *softwares* de virtualização, que emulam equipamentos reais.
- **Equipamentos em modo gráfico:** A possibilidade de executar equipamentos em modo gráfico para navegação entre o sistema operacional do equipamento.
- **Configuração Real das Interfaces de Rede:** Após a devida configuração física simulada pelo ambiente, a configuração real, leva em conta trechos de código, para a configuração das interfaces de rede, que se aproximam da realidade.

- **Captura dos Dados Trafegados:** Aplicação que monitora e captura a comunicação entre equipamentos, facilitando a compreensão dos protocolos e mensagens enviadas entre os equipamentos utilizando a ferramenta Wireshark.
- **Interface de Criação:** Ambiente em que se habilita a possibilidade de criação do ambientes, arrastando elementos, e pré-configurando as suas conexões visualmente.
- **Ambiente de Fácil Execução:** Momento em que o laboratório de estudo já está devidamente configurado e somente aguarda inicialização dos equipamentos.
- **Ambiente de Fácil Criação:** Considera a dificuldade da criação de equipamentos e suas conexões, utilizando um novo ambiente.
- **Código Aberto:** Simuladores que possuem o código disponibilizado de forma livre.

Quadro 1 – Comparativo De Simuladores.

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

3 FERRAMENTAS

Neste capítulo serão descritos os referenciais teóricos que orientam o andamento e implementação do projeto. Apresentando as ferramentas utilizadas, para concretização do *software*.

3.1 Python

A linguagem de programação Python é uma linguagem considerada de alto nível, por conta de sua simples sintaxe e velocidade de desenvolvimento. 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 XBNet faz o uso desta linguagem, para a inicialização dos equipamentos de forma automatizada ([DOWNEY, 2012](#)).

3.1.1 TKinter

O TKinter é uma biblioteca nativa que já vem integrada a projetos que utilizam Python, aplicada ao desenvolvimento de interfaces gráficas ou GUI *Graphic User Interface*, 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?](#)).

3.2 Metodologia Scrum

O *framework Scrum*, é uma ferramenta de gerenciamento de processos, para a realização de algo, tem sido empregado desde o início dos anos 90, o *Scrum* deixa definido abertamente que a utilização de processos e técnicas, fica sob responsabilidade dos envolvidos, de forma não sistemática e tradicional, mas sim otimizada para cada projeto. O coração do *Scrum* possui a intitulação de *Sprint*, que seria uma parcela do produto, desenvolvida em um período de tempo, que mesmo não finalizando o produto, é utilizável e incrementa valor.

3.2.1 Papéis da Equipe Scrum

Product owner: Figura que possui grande conhecimento sobre o negócio, ou que possui forte ligação com o cliente final, sabendo das suas necessidades e dificuldades a serem resolvidas para o melhorar o produto.

Development team: Como a própria tradução literal diz, representa os membros da equipe que realizarão o desenvolvimento do projeto para a geração do produto.

Scrum master: É a figura que geralmente possui um bom conhecimento da equipe, e por conta disso estimula e cobra a teoria do *Scrum* dentro da mesma, mantendo a linha organizacional entre os envolvidos.

3.2.2 Artefatos do Scrum

Product Backlog: É uma lista ordenada de todas as necessidades apontadas pelo *product owner*, que devem ser desenvolvidas pela equipe.

Sprint Backlog: São um conjunto de itens, selecionados do *product backlog*, para a finalização de uma *Sprint*.

3.2.3 Eventos do Scrum

Sprint Planning: É a reunião realizada para o planejamento da primeira ou próxima *Sprint*, definindo a parcela do produto que pode ser entregue no período de tempo definido pela equipe, geralmente em 15 dias ou em 1 mês.

Sprint Review: Reúne a equipe, para a troca de informação sobre os incrementos realizados na *Sprint* finalizada, promove a discussão e melhoramento das soluções apresentadas. (SCHWABER; SUTHERLAND, 2016).

3.2.4 Uso do Scrum No Projeto

Para adaptação do uso do *framework*, cada *Sprint*, fica definida com o período de 15 dias, que são antecedidos de uma reunião de planejamento para definição do desenvolvimento, e escolha da reunião para revisão no período intermediário da *Sprint*, em que os envolvidos, aluno e professor, possam conversar sobre o andamento da solução. O *product backlog* é definido pelas histórias do usuário.

3.3 Trello

Para a realização de gerência das atividades a serem desenvolvidas será utilizado o sistema Trello, nele serão adicionadas sprints que são organizadas visualmente em blocos, utilizando o método Kanban de gerência de projetos, nesse método o projeto pode ser dividido em

grupos e esses grupos podem ser divididos em tarefas com a finalidade de melhor compreensão das atividades (JOHNSON, 2017).

3.4 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.

3.5 Considerações

A aplicação será desenvolvida, utilizando 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.

4 DESENVOLVIMENTO PRELIMINAR

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

4.1 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.

4.1.1 Processo de Criação de Ambientes XBNet

A ilustração a seguir, 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.

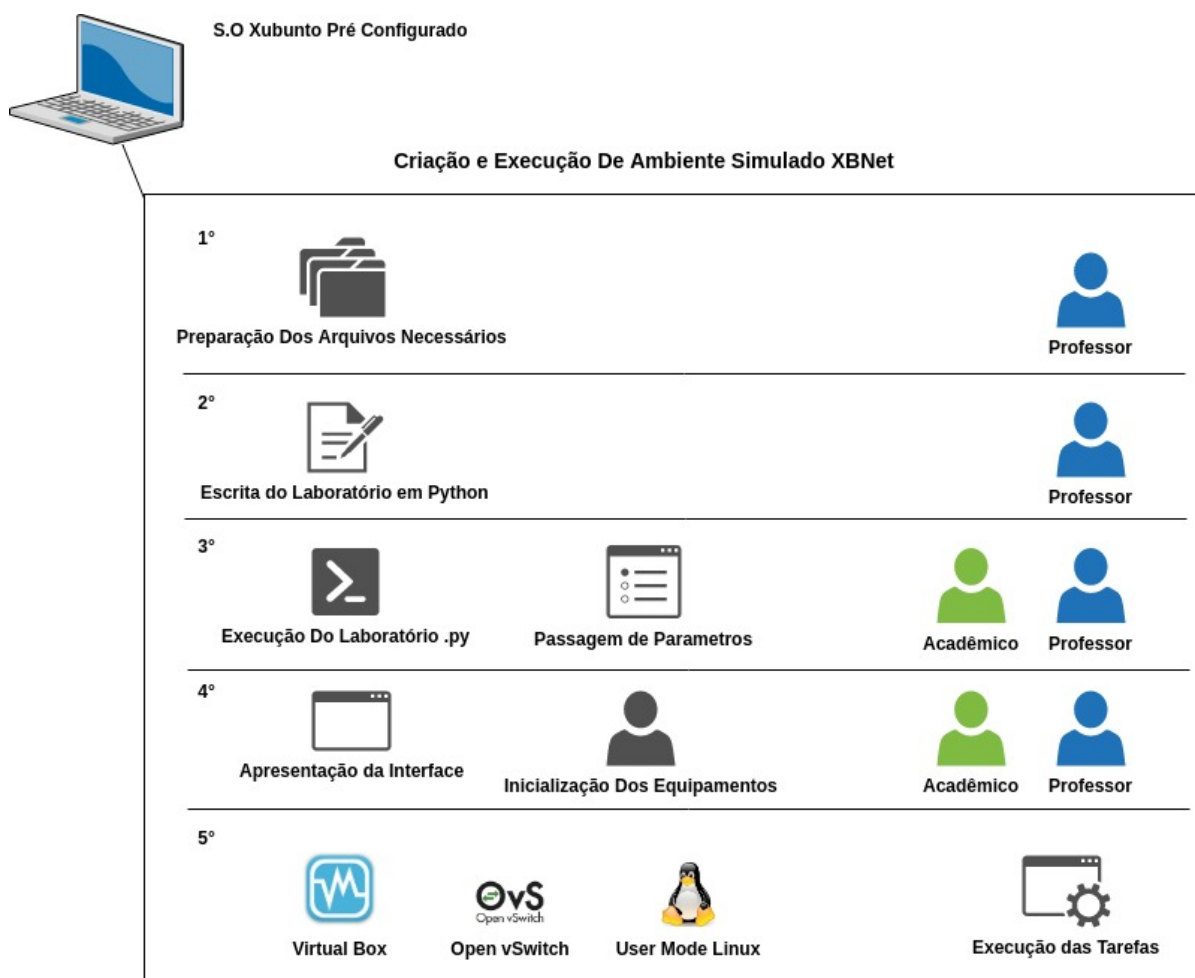
Após a realização da configuração, é possível aluno e professor executar o código fonte Python implementado, inserindo os parâmetros solicitados para criação do ambiente.

Durante o processo quatro, 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.

Figura 6 – Processo de Criação de Ambientes XBNet



Fonte: Autor

4.1.2 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.

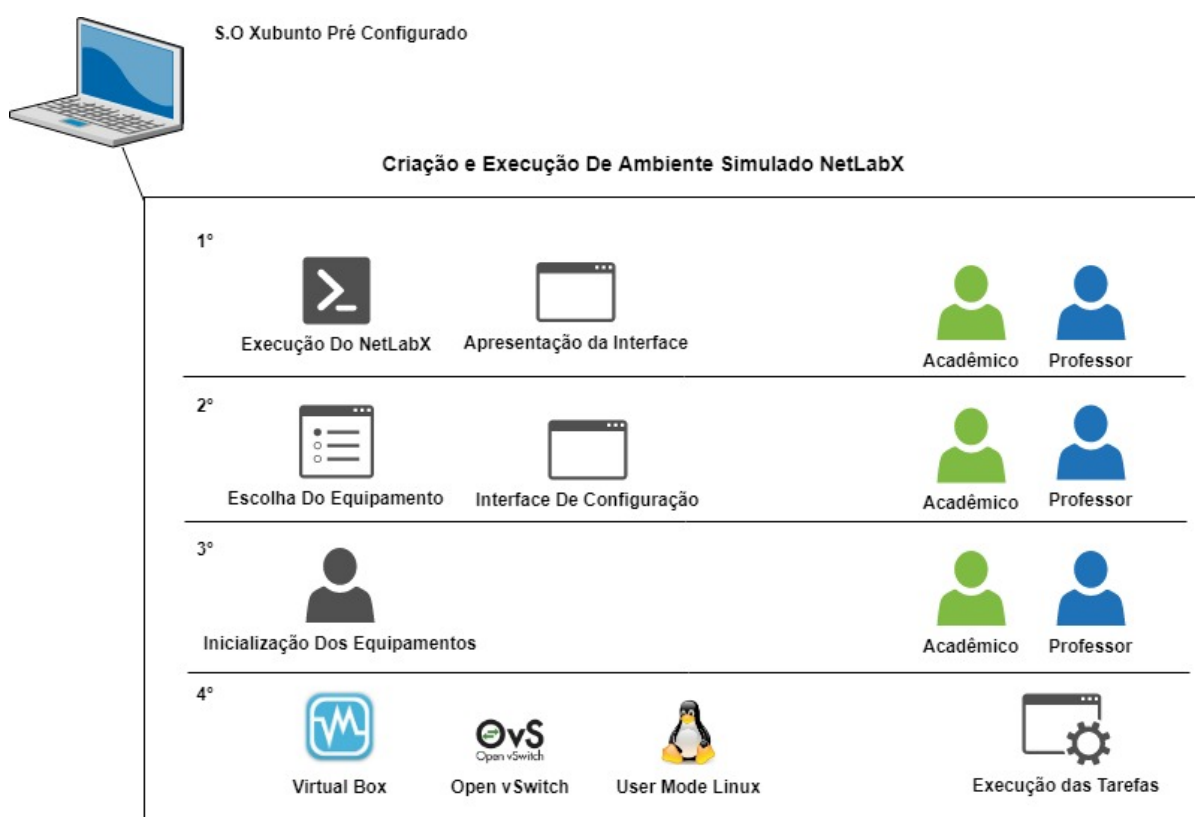
O processo seguinte, ilustra os equipamentos que podem ser criados no ambiente simulado, e realiza uma tela de pré configuração, para relacionar equipamentos e suas conexões necessárias, simulando a forma física do ambiente.

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.

Figura 7 – Proposta de Processo Para Criação de Ambientes NetLabX



Fonte: Autor

4.1.3 Requisitos da Ferramenta Visual NetLabX

Sendo assim, justamente o objeto desse projeto, é a interface de comunicação entre o usuário, e o código de inicialização de equipamentos, omitindo sua programação, gerando um ambiente pronto para o desenvolvimento de atividades e ou estudo.

Em conjunto com o professor, que 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 Quadro 2, onde são definidas as ações quais, o usuário gostaria de realizar durante a utilização da aplicação proposta.

Quadro 2 – Histórias do Usuário.

História	Descrição
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.
H8	Como usuário, eu gostaria de visualizar os elementos que estão ligados e desligados.

4.2 PROJETO DO SISTEMA

Nesta seção são abordados, os artefatos, que devem ser desenvolvidos para a elaboração e sucesso do projeto de sistema.

Serão apresentadas, as decisões arquiteturais, os protótipos de tela para o sistema, assim como a descrição de suas funcionalidades. E logo após a apresentação do diagrama de classes, para visualização das questões de implementação do projeto.

4.2.1 Decisões Arquiteturais

Python: A utilização da linguagem Python, foi selecionada, relacionando os elementos já desenvolvidos pelo XBNet, 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.

Tkinter: 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 é considerada complexa, o uso da ferramenta nativa Python, simplifica a implementação.

Geração de Arquivos: A geração de arquivos, ainda deverá passar por processo de pesquisa, para sua correta implementação, a aplicação deve compilar as informações salvas convertendo os elementos visuais do laboratório em memória, para linhas de texto em arquivos, utilizando JSON *JavaScript Object Notation*, posteriormente, durante a abertura desses arquivos dos projetos, todas essas linhas devem ser interpretadas, para a conversão desses objetos armazenados, em instâncias na memória novamente, e assim carregando todos os equipamentos configurados visualmente, possibilitando a sua execução.

4.2.2 Protótipo das Telas do Sistema

Para a elaboração do protótipo de telas, foi realizada a utilização da ferramenta on-line draw.io. Com essa ferramenta foi possível organizar a *interface*, com as histórias do usuário, atendendo assim a necessidade apresentada. A produção dessas telas durante o desenvolvimento, podem não seguir fielmente os modelos apresentados, mas podem expressar como a ferramenta deve funcionar em seu uso, obtendo uma prévia, e se caso houver necessidade de remodelagem ou correção, não utilizando tempo demais, somente para sua visualização de tela.

4.2.3 Tela Principal

Observando a Figura 8, se tem a representação da 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.

Ao centro a visualização primordial para a fácil manipulação dos equipamentos, onde 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 *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.

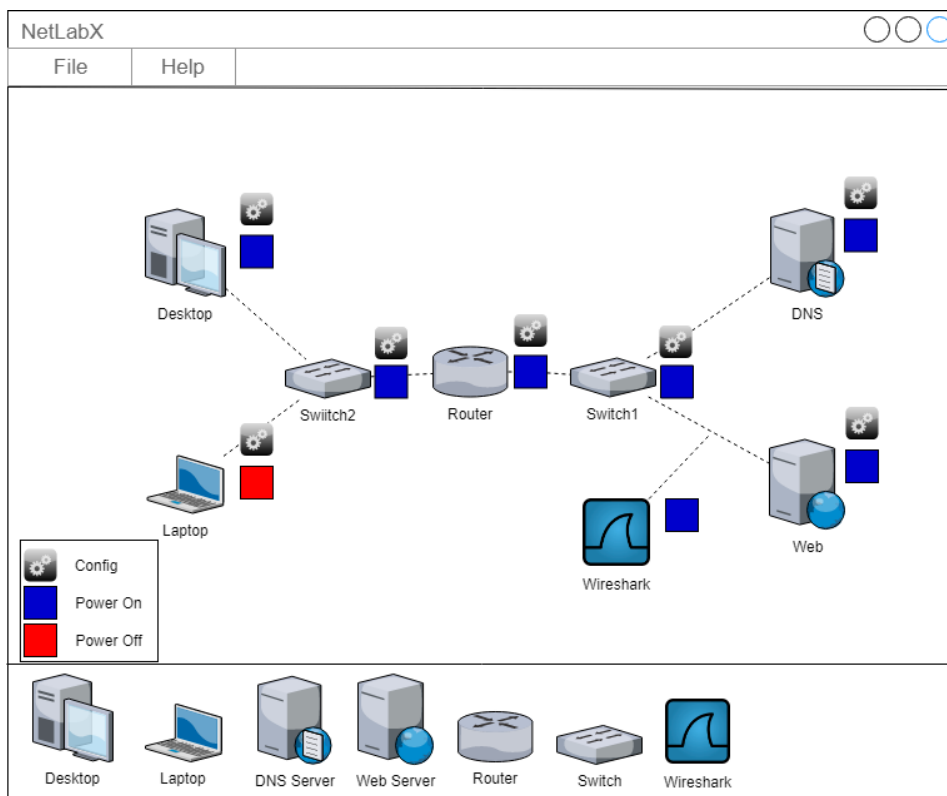
Esses equipamentos, quando executados, geram uma nova tela que é semelhante a gerada em um equipamento físico, que possui um sistema operacional Linux, podendo gerar em modo gráfico, onde o usuário utiliza teclado e *mouse*, para navegar entre as telas do sistema, ou a opção de *console*, onde o sistema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

A visualização energética do equipamento, é simulada com um pequeno quadro que ilustra sua condição de desligado ou ligado. Ao ser clicado essa ilustração altera sua cor, e invoca a criação desse equipamento, abrindo assim uma janela de *console* ou de modo gráfico para sua utilização.

Na ilustração seguinte, são listados os equipamentos disponíveis para criação do ambiente, cada equipamento, ao ser clicado, aciona a tela de configuração, e quando salvo é adicionado ao projeto do laboratório, sendo visualizado ao centro da tela, para que o usuário

possa move-lo até a posição que melhor desejar.

Figura 8 – Protótipo da Tela Principal



Fonte: Autor

4.2.4 Tela 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, e também conexões, salvando essa configuração, o equipamento é criado e já pode ser inicializado, ficando a cargo do usuário fechar equipamentos que podem ter sido inicializados anteriormente a configuração, fazendo com que os elementos novos, não se comuniquem com os já criados em nível de execução.

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 da Tela de Configuração do Equipamento

O protótipo da tela de configuração do equipamento, intitulada "Config Equipment", é exibida dentro de uma janela do NetLabX. A interface possui um menu superior com "File" e "Help". O formulário principal 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 "ip" e "gateway", e dois menus suspensos "Interface" e "Equipment", seguidos por um botão "Add".
- Gráficos de Modo:** Três controles deslizantes para "Graphical Mode", "Console" e "DHCP", todos configurados para "ON".
- Botão de Ação:** Um botão "Save" localizado no canto inferior direito.
- Barra de Navegação:** Uma barra na base da janela com ícones e rótulos para "Desktop", "Laptop", "DNS Server", "Web Server", "Router", "Switch" e "Wireshark".

Fonte: Autor

4.2.5 Projeto Diagrama de classes

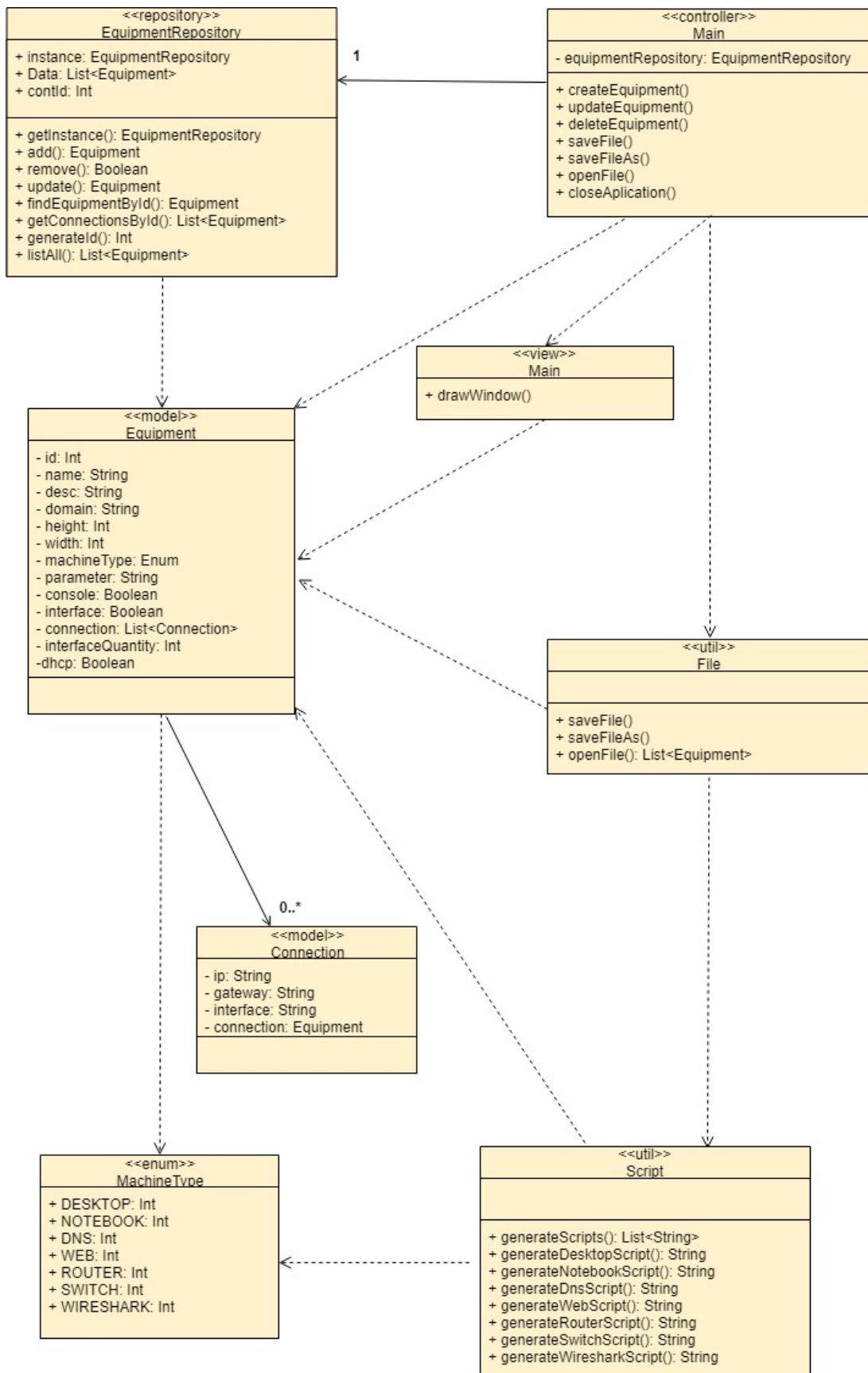
Obtendo a necessidade de salvar projetos criados na aplicação em arquivos, para o desenvolvimento se fez desnecessário a utilização de base de dados, por conta de que, encerrando a aplicação, os dados registrados e manipulados, não são mais necessários para o correto funcionamento.

Com isso a aplicação será executada diretamente em memória, onde segundo o diagrama de classes, o acesso a informação manipulada pela aplicação, é realizada através de uma classe utilizando o padrão de projeto *Singleton*, com isso é possível garantir que somente uma instancia deste objeto é realizada, essa classe simula o comportamento de uma conexão de base de dados, onde toda a informação referente a objetos criados, deve estar contida nesse objeto.

Informações, referentes a equipamentos que serão criados, como computadores e servidores, poderão ser referenciados em arquivos de configuração, arquivos estes que são executados durante a abertura da aplicação.

A Figura 10, representa a implementação que deverá ser realizada, para que o projeto possua todas as ações e elementos de representação, solucionando os problemas apresentados.

Figura 10 – Diagrama de classes



Fonte: Autor

4.3 PLANEJAMENTO DO TRABALHO

O Quadro 3, apresenta o cronograma das atividades, estimadas para o desenvolvimento do projeto.

Quadro 3 – Cronograma de Atividades.

Atividades	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1. Revisão dos apontamentos da banca		X								
2. Revisão bibliográfica		X								
3. Requisitos do Projeto			X							
4. Redação do Projeto de TCC		X	X	X	X					
5. Diagramação e Protótipos de tela					X	X				
6. Início do Desenvolvimento					X	X	X	X	X	
7. Construção de Telas					X	X				
8. Integração Com as Configurações de Virtualização						X	X	X		
9. Escrita da Monografia de TCC						X	X	X	X	
10. Elaboração da apresentação final								X	X	
11. Defesa final do TCC									X	

4.4 CONSIDERAÇÕES FINAIS

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.

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. Citado na página 13.
- BOMBAL, D.; DUPONCHELLE, J. **Getting Started with GNS3**. 2019. Disponível em: <https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.html>. Acesso em: 15 de abril de 2019. Citado na página 7.
- BORGES, V.; SARAIVA, O.; CAMARGO, L. **L.S.de A. de. Software simuladores de rede: análise comparativa para apresentação de funcionalidades e benefícios**. in: lii simtec – simpósio de tecnologia da fatec taquaritinga. 2015. Citado 2 vezes nas páginas 1 e 5.
- 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>. Acesso em: 17 de abril de 2019. Citado na página 6.
- 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. Citado na página 11.
- JOHNSON, H. **Trello**: Journal of the medical library association. 2017. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5370621/>>. Acesso em: 16 de abril de 2019. Citado na página 13.
- 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. Citado na página 11.
- LODDO, J.-V.; SAIU, L. Marionnet: a virtual network laboratory and simulation tool. p. 1–3, 2008. Citado na página 8.
- RIMONDINI, M. Emulation of computer networks with netkit. p. 3–5, 2007. Citado na página 5.
- SCHWABER, K.; SUTHERLAND, J. **The scrum guide - the definitive guide to scrum: The rules of the game**. 2016. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf>>. Acesso em: 16 de abril de 2019. Citado na página 12.