

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

PATRICK FERRO RIBEIRO

**NETLABX - INTERFACE PARA CONTROLE DE AMBIENTE
VIRTUALIZADO DE REDE**

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2019

PATRICK FERRO RIBEIRO

**NETLABX - INTERFACE PARA CONTROLE DE AMBIENTE
VIRTUALIZADO DE REDE**

Proposta 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

1 PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

1.1 TÍTULO

NetLabx - Interface Para Controle de Ambiente Virtualizado de Rede

1.2 MODALIDADE DO TRABALHO

Desenvolvimento de Sistemas

1.3 ÁREA DO TRABALHO

Desenvolvimento de sistema desktop.

1.4 RESUMO

Com o crescimento da rede mundial de computadores e a sua utilização, torna-se necessário estudos e abstrações de conceitos utilizados na área. Desta forma surge grande demanda por aplicações relacionadas a segurança, expansão e seu manutenção.

A definição de redes de computadores se da pelo conjunto de equipamentos interligados de maneira a trocaram informações e compartilharem 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. Como citado anteriormente, um ambiente simulado necessita quantidade de equipamentos físicos reduzidos, e sua configuração pode ser realizada de forma automatizada. Em atividades realizadas em sala, constatou-se a carência de um ambiente visualmente mais intuitivo e de simples configuração, onde qualquer acadêmico ou professor possa realizar suas simulações e testes, sem demandar tempo excedente na criação desse ambiente.

2 DESCRIÇÃO DA PROPOSTA

2.1 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 precisa ser muito abrangente, ao longo do trabalho serão citados especificamente as atividades de trabalho de um profissional de redes, como por exemplo levantamento de custos, facilidade de acesso, padronização, segurança, portabilidade e configuração de equipamentos. Todas essas atividades, 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 complexabilidade 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 a eles. 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).

2.2 OBJETIVOS

2.2.1 OBJETIVO GERAL

Desenvolvimento de interface gráfica, para o gerenciamento de equipamentos virtualizados. Simplificando o processo da criação dos laboratórios de rede, utilizados em sala de aula.

2.2.2 OBJETIVOS ESPECÍFICOS

São objetivos específicos desse trabalho:

- Realizar a comunicação com a virtualização, sem necessidade de programação pelo usuário;
- Ilustrar os equipamentos de rede utilizados, e possibilitar a gerencia dos comportamentos energéticos, como ligar e desligar.
- Salvar e exportar configurações do ambiente criado, para utilização em outros equipamentos.
- Definir ambientes pré instalados e configurados, facilitando a execução de atividades práticas;

2.3 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) Campus Guarapuava, é intitulado de XBNet que possui esse nome por conta de ser uma distribuição Xubuntu modificada com finalidade de trabalhar em sincronia 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 qual, comunica 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 destes ambientes requerem conhecimentos de programação, visto que para criar novos ambientes é necessária a implementação de código programático.

2.4 ESTADO DA ARTE

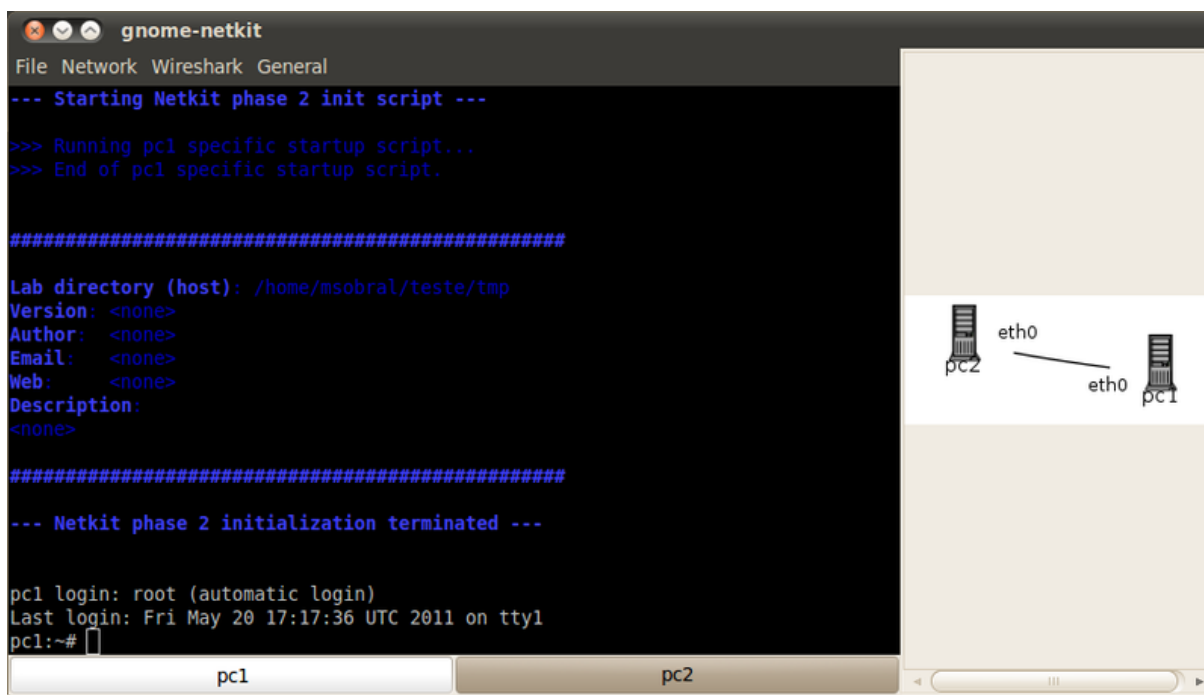
Nesta seção são apresentadas tecnologias que existem e estão disponíveis para seu uso através de licenças de software distintas, buscando soluções para problemas que estão relacionados ao objetivo deste projeto.

2.4.1 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). foi desenvolvido sob os termos da licença de uso GPL(General Public Licence) e utiliza software Livre(Open Source), tem como principio 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 1 – Captura de Tela NetKit

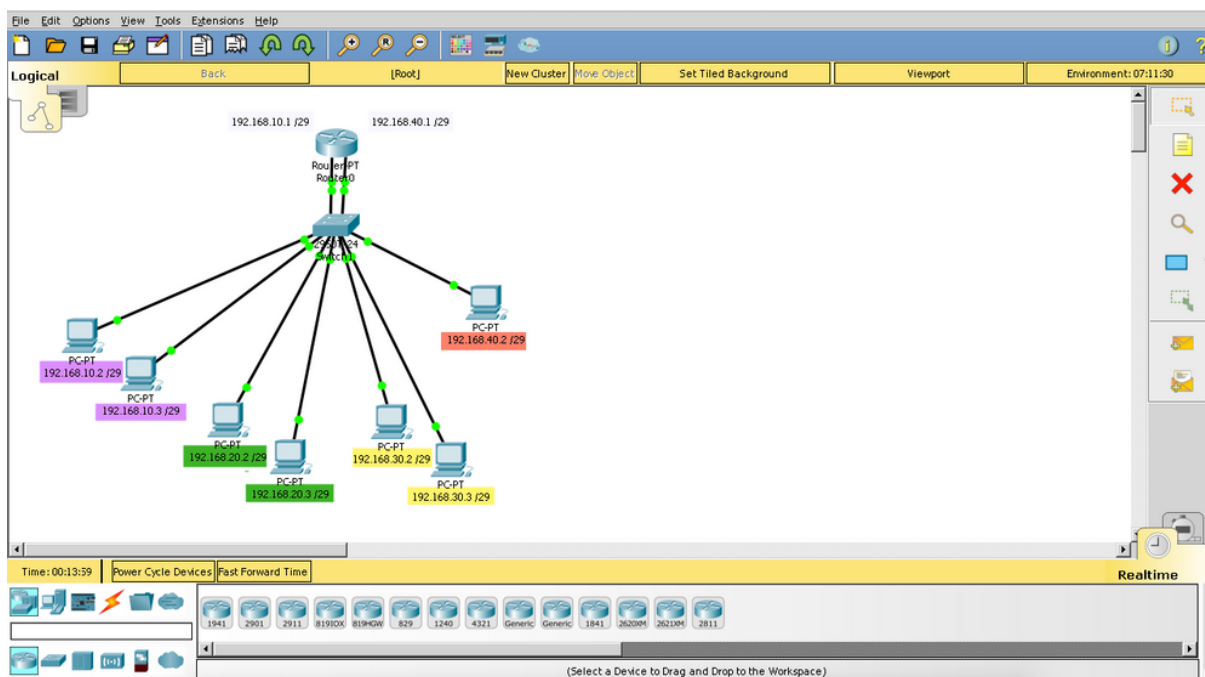


Fonte: <https://wiki.sj.ifsc.edu.br/wiki/index.php/Netkit2>

2.4.2 Cisco Packet Tracer

O software de simulação de ambientes de rede chamado Cisco Packet Tracer foi desenvolvido pela empresa Cisco 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 2 – Captura de Tela Cisco Packet Tracer



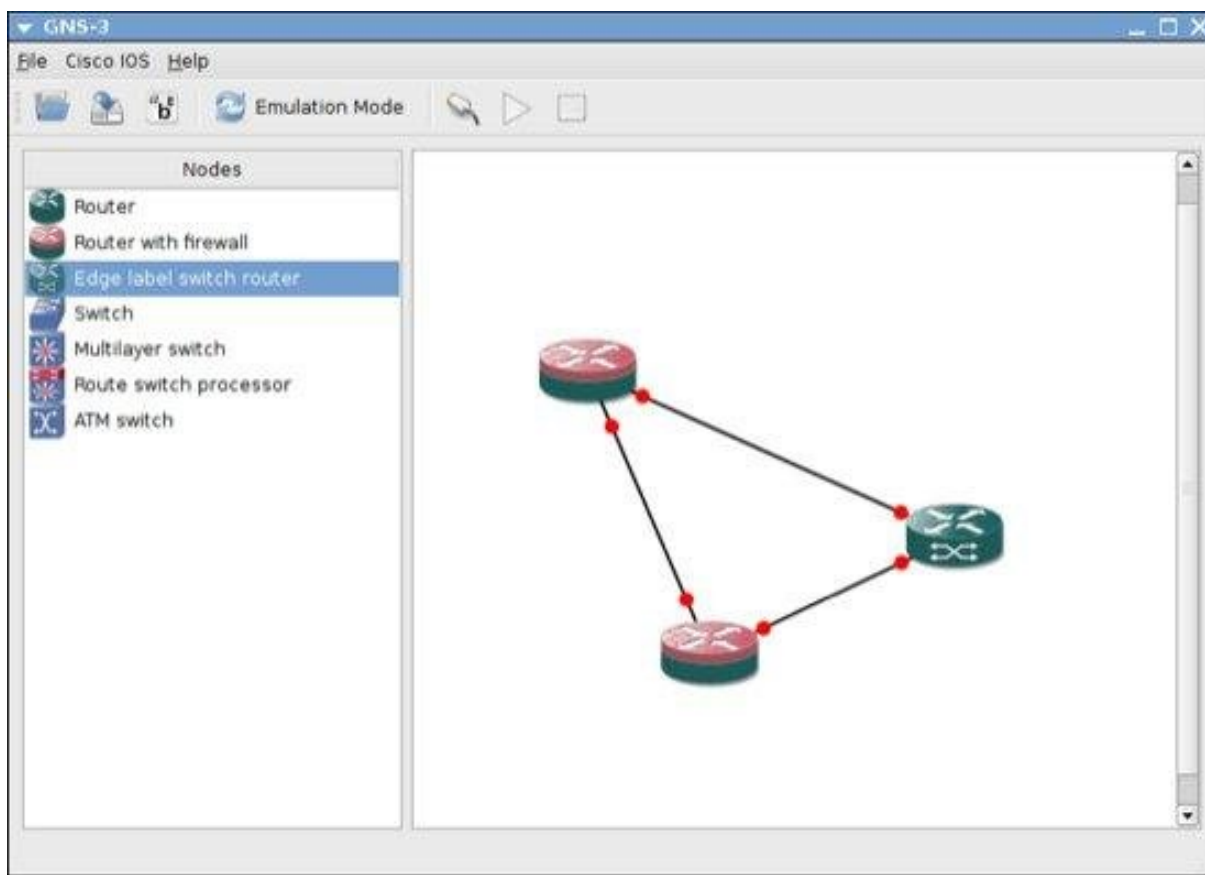
Fonte:

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

2.4.3 GNS3

É um software livre e possui código aberto para a comunidade, a ferramenta possui as funcionalidades de simulação e emulação, e 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 3 – Captura de Tela GNS3

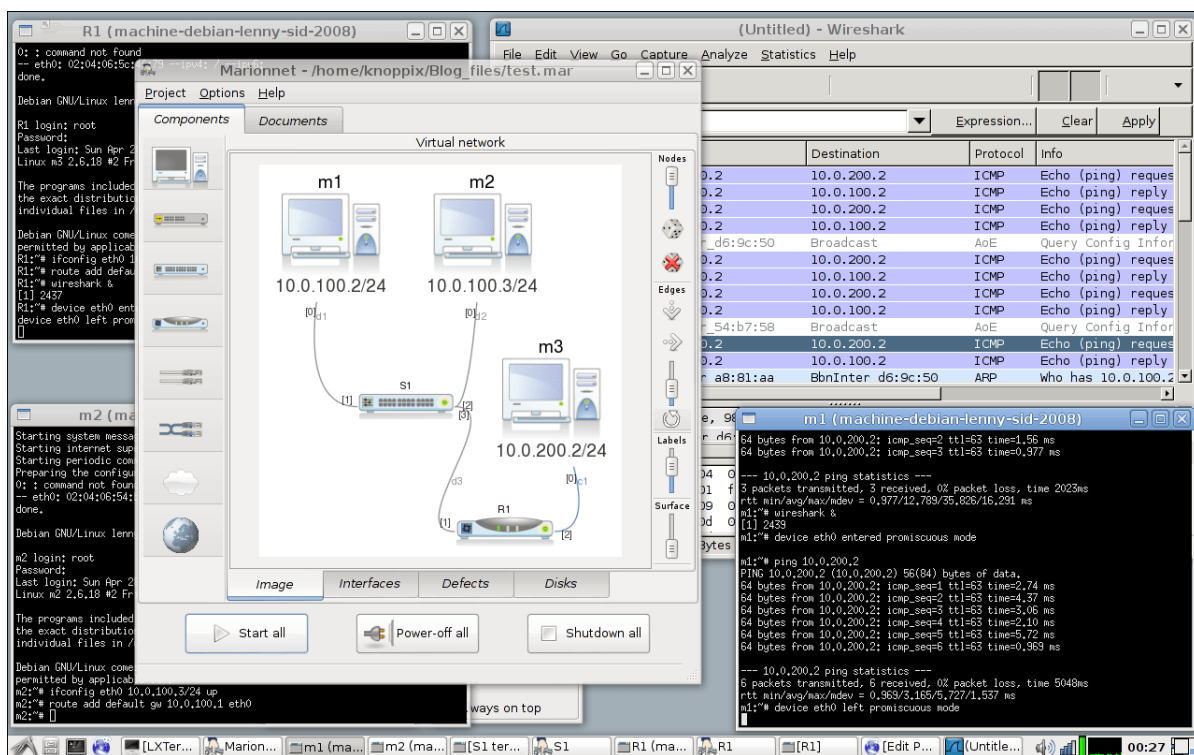


Fonte: <https://sourceforge.net/projects/gns-3/>

2.4.4 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, e a configuração é feita através do terminal do sistema, que executam 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 totalmente atualizado e configurado com as dependências necessárias, e mesmo assim podem ocorrer erros durante a execução (LODDO; SAIU, 2008).

Figura 4 – Captura de Tela Marionnet



Fonte: <https://www.brianlinkletter.com/marionnet-network-simulator-test-drive/>

2.5 PROCEDIMENTOS METODOLÓGICOS/METODOLOGIA

Primariamente ao desenvolvimento, será necessário a pesquisa para utilizar as ferramentas que possuam melhores recursos, durante as atividades de criação da interface, pois existe a necessidade de interface gráfica intuitiva, onde o investimento de tempo em uma tecnologia demasiada complicada, pode tornar o projeto complexo, e comprometer as telas do software.

Após a definição da tecnologia abordada serão realizadas as modelagens do sistema, que definirão em forma de diagramas a comunicação entre os elementos e os primeiros protótipos de telas.

O desenvolvimento do sistema seguirá os princípios básicos da metodologia Scrum de desenvolvimento ágil, efetuando algumas adaptações por conta de que o projeto não possui todos os papéis necessários determinados no Scrum, onde juntamente com o professor orientador, durante a reunião de orientação, serão levantadas informações sobre o andamento do projeto, e o apontamento de necessidades ou alterações que possam surgir na resolução da problemática, onde por sua vez essas tarefas ou atividades são divididas em parcelas tituladas de sprints, que possuem um período de tempo para serem realizadas, no caso do desenvolvimento, cada sprint fica definida em uma semana, para que aluno e orientador tenham o tempo necessário de pesquisa e solução das atividades listadas, apresentando na reunião da semana posterior as dificuldades e soluções encontradas (SCHWABER; SUTHERLAND, 2016).

Para a realização de gerencia 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 gerencia 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).

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).

Ainda durante o desenvolvimento serão liberadas pequenas atividades da aplicação com o intuito de testes de usabilidade.

Serão realizados os ajustes necessários baseados nas informações levantadas durante todo a fase de desenvolvimento, então por fim a realização de entrega do software a comunidade acadêmica.

2.6 CONSIDERAÇÕES FINAIS

Com a obtenção das informações referentes a softwares já existentes, pode-se visualizar, a carência de um projeto que seja desenvolvido buscando a interação e exibição completa de um ambiente de rede, onde cada elemento possa ser simulado, configurado e apresentado de forma que um estudante neste ambiente virtual, possa obter conhecimento que será utilizado no ambiente real, e além disso não seja necessário o entendimento em outras áreas. Onde será possível a execução de projetos básicos, como o exemplo da rede domestica citado anteriormente, que poderá ser modificado pelo acadêmico ou professor, expandindo as possibilidades do objeto de estudo.

2.7 PLANEJAMENTO DO TRABALHO

O planejamento do trabalho que será desenvolvido pelo aluno, ao longo do período letivo, está descrito no cronograma da Quadro 1. Neste cronograma constam todas as atividades com seus respectivos prazos para o cumprimento.

Quadro 1 – 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	X	X	X					
3. Requisitos do Desenvolvimento			X							
4. Diagramação e Protótipos de tela			X							
5. Início do Desenvolvimento			X	X						
6. Redação do projeto de TCC			X	X						
7. Criação de Artes e Objetos				X						
8. Criação de Telas Funcionais				X	X	X				
9. Integração Com os Arquivos de Virtualização				X	X	X	X	X		
10. Escrita da Monografia de TCC						X	X	X		
11. Elaboração da apresentação final								X	X	
12. Defesa final do TCC									X	

2.7.1 DA PROPOSTA DO PROJETO

Durante o período posterior a aprovação da proposta e entrega do projeto pretende-se definir os seguintes pontos:

1. Definir a tecnologia a ser utilizada, para criação da aplicação desktop.
2. Definir as principais melhorias do software.
3. Definir as telas a serem desenvolvidas.
4. Apresentar as etapas e problemas encontrados na resolução do projeto.
5. Apresentar os equipamentos que poderão ser utilizados na ferramenta.
6. Apresentar a comparação com os softwares relacionados.

Assim, estes itens estarão descritos no projeto de TCC.

2.8 RECURSOS NECESSÁRIOS

- Equipamentos que utilizam sistema operacional de distribuição livre.
- Acesso ao site Trello, para gerencia das atividades.
- Acesso ao GitHub para gestão de versão do código online.

Todos os recursos são de total responsabilidade do acadêmico.

2.9 HORÁRIO DE TRABALHO

O horário destinado para realização das atividades do TCC, bem como o horário destinado para a reunião semanal com o orientador estão descritos no cronograma do Quadro 2. Este horário é definido com orientador levando em consideração a complexidade do trabalho a ser desenvolvido.

Quadro 2 – Horário de Trabalho.

Horário	Seg	Ter	Qua	Qui	Sex	Sab
09h30 - 10h30						TCC
10h30 - 11h30						TCC
13h00 - 13h50						TCC
13h50 - 14h40						TCC
14h40 - 15h30						TCC
17h00 - 18h00	TCC	TCC	TCC	TCC	TCC	
19h30 - 20h30	orientação			orientação		
20h30 - 21h20	TCC	TCC	TCC			
21h30 - 22h15	TCC	TCC	TCC			

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 8.
- 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 5.
- 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 2 e 4.
- 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 4.
- 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 8.
- LODDO, J.-V.; SAIU, L. Marionnet: a virtual network laboratory and simulation tool. p. 1–3, 2008. Citado na página 6.
- RIMONDINI, M. Emulation of computer networks with netkit. p. 3–5, 2007. Citado na página 3.
- 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 7.