

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

CLEITON MIGLIORINI

**AMBIENTE DE ACESSO VIRTUAL PARA AUXÍLIO NO ENSINO DE
REDES DE COMPUTADORES.**

MONOGRAFIA DE ESPECIALIZAÇÃO

PATO BRANCO
2018

CLEITON MIGLIORINI

**AMBIENTE DE ACESSO VIRTUAL PARA AUXÍLIO NO ENSINO DE
REDES DE COMPUTADORES.**

Monografia de especialização apresentada ao III Curso de Especialização em Redes de Computadores - Configuração e Gerenciamento de Servidores e Equipamentos de Rede, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Fabiano Scriptori de Carvalho, Msc.

PATO BRANCO
2018

TERMO DE APROVAÇÃO

AMBIENTE DE ACESSO VIRTUAL PARA AUXÍLIO NO ENSINO DE REDES DE COMPUTADORES

por

Cleiton Migliorini

Esta monografia foi apresentada às 08h00min do dia 23 de novembro de 2018, como requisito parcial para obtenção do título de ESPECIALISTA, no III Curso de Especialização em Redes de Computadores - Configuração e Gerenciamento de Servidores e Equipamentos de Redes, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho **aprovado**.

Prof. Me. Fabiano Scriptore de
Carvalho
Orientador / UTFPR-CT

Prof. Dr. Fábio Favarim
UTFPR-PB

Prof. Dr. Eden Ricardo Dosciatti
UTFPR-PB

Prof. Dr. Fábio Favarim
Coordenador do III Curso de Especialização
em Redes de Computadores

Dedico este trabalho à minha família.

Pouco conhecimento faz com que as criaturas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o céu, enquanto que as cheias a baixam para a terra, sua mãe.

Leonardo da Vinci

RESUMO

MIGLIORINI, Cleiton Migliorini. Ambiente de acesso virtual para auxílio no ensino de rede de computadores. 2018. 71f. Monografia (Especialização em Redes de Computadores) - Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Campus Pato Branco. Pato Branco, 2018.

O projeto tem como objetivo o desenvolvimento de uma plataforma web para a realização da configuração e acesso remoto a equipamento de redes, com o objetivo de apoiar as aulas práticas de disciplinas que envolvem redes de computadores e telecomunicação, buscando facilitar o aprendizado nestas áreas com o emprego de uma abordagem em equipamentos reais. O software elaborado por este projeto busca também realizar um controle de equipamentos e seu melhor aproveitamento em aulas ou projetos. O desenvolvimento do projeto foi realizado em Python com a utilização do framework Flask, para a parte do cliente foi utilizado Angular 6, HTML e CSS, o que possibilita o acesso do usuário em qualquer navegador e em qualquer lugar.

Palavras-chave: Acesso remoto, web, Redes de Computadores, Python e Angular.

ABSTRACT

MIGLIORINI, Cleiton Migliorini. Virtual access environment to aid in the teaching of computer networks. 2018. 71f. Monografia (Especialização em Redes de Computadores) - Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

The project aims to develop a web platform for the configuration and remote access to network equipment, with the objective of supporting the practical classes of disciplines that involve computer networks and telecommunication, seeking to facilitate the learning in these areas with the use of a real equipment approach. The software developed in this project also seeks to perform a control of equipment and its best use in classes or projects. The development of the project was carried out in python using the Flask framework, for the part of the client was used Angular 6, HTML and CSS, which allows the user access in any browser and place.

Keywords: Web Platforms, Remote Access, Networks, Python e Angular.

LISTA DE FIGURAS

Figura 1 - Rede ponto a ponto.....	18
Figura 2 - Rede Multiponto.....	19
Figura 3 - Modelos de referência.....	23
Figura 4 - Placa de rede.....	24
Figura 5 - Simulador de voo.....	27
Figura 6 - Tela do GNS3.....	28
Figura 7 - Tela do Packet Tracer.....	29
Figura 8 - Tela do NETLAB CISCO.....	30
Figura 9 - Diagrama de classe estrutural.....	34
Figura 10 - Diagrama de caso de uso.....	34
Figura 11 - Estrutura básica do angular.....	41
Figura 12 - Tela inicial do Astah.....	42
Figura 13 - switch Cisco 2960.....	43
Figura 14 - switch L3 3750.....	43
Figura 15 - Roteador Cisco 2811.....	44
Figura 16 -Roteador Cisco 3845.....	44
Figura 17 - Diagrama de caso de sistema.....	47
Figura 18 - Diagrama de classe do sistema.....	47
Figura 19 - Protótipo de tela principal.....	52
Figura 20 - Protótipo de telas de cadastro, consultas e exclusões.....	53
Figura 21 - Protótipo de tela de cadastro.....	53
Figura 22 - Protótipo da tela de cadastro de reservas.....	54
Figura 23 - Tela de visualização de reservas.....	55
Figura 24 - Protótipo de tela de acesso via terminal.....	55
Figura 25 - Tela de login.....	56
Figura 26 - Tela inicial do sistema.....	57
Figura 27 - Tela de consulta de usuário.....	59
Figura 28 - Tela de cadastro de usuário.....	60
Figura 29 - Tela de consulta de equipamento.....	61
Figura 30 - Tela de cadastro de equipamento.....	62
Figura 31 - Tela de reserva.....	62
Figura 32 - Tela de consulta de reserva por parte do aluno.....	63
Figura 33 - Tela para interação com o equipamento.....	64

LISTA DE QUADROS

Quadro 1 - Descrição de campos de usuários.....	49
Quadro 2 - Descrição de campos de turmas.....	50
Quadro 3 - Descrição de campos dos equipamentos.....	50
Quadro 4 - Descrição dos campos da reserva.....	51

LISTA DE CÓDIGOS

Listagem 1 - Definição de rotas.....	58
Listagem 2 - Chamada de rotas no HTML.....	58
Listagem 3 - Requisição para API em python.....	59
Listagem 4 - API em python buscando relacionar a busca dos usuários.....	60
Listagem 5 - Exemplo de comunicação serial realizada com o equipamento.	64

LISTA DE CÓDIGOS

Tabela 1 - Tabela de versões.....	37
-----------------------------------	----

LISTA DE SIGLAS

CSS3	Cascading Style Sheets
HTTP	<i>Hypertext Transfer Protocol</i>
HTML	<i>HyperText Markup Language</i>
IP	<i>Internet Protocol</i>
IU	Interface de usuário
ISO	<i>International Organization for Standardization</i>
LANs	<i>Local Area Network</i>
NIC	<i>Network Interface Card</i>
OSI	<i>Open Systems Interconnection</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
UTFPR	<i>Universidade Tecnológica Federal do Paraná.</i>
XML	<i>Extensible Markup Language</i>
WANS	<i>Wide Area Network</i>

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	OBJETIVOS.....	15
1.1.1	<i>Objetivo Geral.....</i>	15
1.1.2	<i>Objetivos Específicos.....</i>	15
1.2	JUSTIFICATIVA.....	16
1.3	ESTRUTURA DO TRABALHO.....	16
2	REFERENCIAL TEÓRICO.....	18
2.1	ORGANIZAÇÃO E FUNCIONAMENTO DE REDE DE COMPUTADORES.....	18
2.2	PROTOCOLO IP.....	20
2.3	MODELO OSI.....	20
2.4	MODELO TCP/IP.....	21
2.5	EQUIPAMENTO DE REDE.....	23
2.5.1	<i>Placa de rede.....</i>	24
2.5.2	<i>Hub.....</i>	24
2.5.3	<i>Switches.....</i>	25
2.5.4	<i>Roteadores.....</i>	26
2.6	SIMULADORES E EMULADORES DE CENÁRIOS DE REDE.....	26
2.6.1	<i>GNS3.....</i>	27
2.6.2	<i>Cisco Packet Tracer.....</i>	29
2.6.3	<i>NETLAB CISCO.....</i>	29
2.7	ENGENHARIA DE SOFTWARE.....	30
2.8	ENGENHARIA DE REQUISITOS.....	31
2.9	ANÁLISE ORIENTADA A OBJETOS (UML).....	32
2.9.1	<i>Diagramas estruturais.....</i>	33
2.9.2	<i>Diagramas comportamentais.....</i>	34
2.10	DESIGN DE INTERAÇÃO.....	34
2.10.1	<i>Prototipação.....</i>	35
2.10.2	<i>Design de interface.....</i>	35
3	MATERIAIS E METODOLOGIA.....	37
3.1	MATERIAIS.....	37
3.1.1	<i>Python.....</i>	37
3.1.2	<i>Flask.....</i>	38
3.1.3	<i>PostgreSQL.....</i>	38
3.1.4	<i>Node.js.....</i>	38
3.1.5	<i>HTML.....</i>	39
3.1.6	<i>Web Services.....</i>	39
3.1.7	<i>Angular.....</i>	40
3.1.8	<i>Astah.....</i>	41
3.1.9	<i>Switch 2960 e Switch L3 3750.....</i>	42
3.1.10	<i>Roteador 2811 e 3800.....</i>	43
3.2	METODOLOGIA.....	44
4	RESULTADOS.....	46
4.1	CRIAÇÃO DO CENÁRIO DE REDE.....	46
4.2	ANÁLISE DA PLATAFORMA VIRTUAL.....	46
4.2.1	<i>Usuários.....</i>	48
4.2.2	<i>Turmas.....</i>	49
4.2.3	<i>Equipamentos.....</i>	50
4.2.4	<i>Reservas.....</i>	50
4.2.5	<i>Prototipação das telas.....</i>	51

4.2.6	<i>Desenvolvimento do sistema</i>	56
5	CONCLUSÕES	66
	REFERÊNCIAS	68

2 INTRODUÇÃO

O sistema de educação mundial está evoluindo a cada dia, isso se deve pelo avanço dos recursos tecnológicos que estão sendo empregados nesta área. Dessa forma o sistema educacional vem caminhado para um rumo de novas possibilidades para o enriquecimento das técnicas de ensino, devendo sempre visar o bom desenvolvimento dos alunos (FERREIRA, LIMA, et al., 2013).

Uma das melhorias no cenário de educação, que está cada dia mais presente, é a utilização de plataformas virtualizadas de ensino. Essas plataformas oferecem ferramentas de apoio aos processos de ensino e aprendizagem, disponibilizando conteúdos e recursos de forma mais eficaz e criando assim uma nova forma de interação entre recursos tecnológicos e usuários (OLIVEIRA, CORTIMIGLIA, LONGHI, 2015).

Esta modalidade de ensino oferece uma forma repleta de interatividade e recursos que podem ser utilizados no sistema educacional, entretanto, isso faz com que a presença e a interação dos acadêmicos com equipamentos disponíveis em laboratórios de informática se torne difícil, para resolver parte destes problemas existem os laboratórios virtuais (FERREIRA, DE LIMA, et al., 2013).

Segundo Prieto-Blázquez, Herrera-Joancomartí, et al., (2009) laboratórios virtuais são espaços interativos que incorporam recursos tecnológicos, seja ele físico ou lógico, com o objetivo da disponibilização de recursos para a realização de atividades práticas, que são adaptadas às necessidades de alunos e professores em uma aprendizagem virtual.

Na sociedade atual, um dos principais recursos tecnológicos que está em constante crescimento, são as redes de comunicação. Com isso, as disciplinas envolvendo esta área, sempre devem buscar levar um aprimoramento e um aperfeiçoamento aos acadêmicos, tornando as atividades práticas essências, no decorrer dos cursos.

Para Tanenbaum (2011), redes de computadores são conjuntos de computadores autônomos constituídos de softwares e hardwares de redes que estão interconectados por algumas tecnologias, proporcionando com

isso uma conexão e troca de informações via recursos físicos e lógicos de uma forma globalizada.

Com base nos conceitos apresentados, o trabalho em questão busca implementar uma plataforma virtual para o auxílio no ensino e aprendizagem de disciplinas que envolvem redes de computadores da Universidade Tecnológica Federal do Paraná (UTFPR). O projeto visa por meio de um módulo de gestão organizar e disponibilizar os recursos físicos dos laboratórios de redes para que docentes e discentes possam ter um acesso mais controlado e uma disponibilidade de recursos.

2.1 OBJETIVOS

2.1.1 Objetivo Geral

Desenvolver um ambiente virtual para auxílio no ensino de redes de computadores que permita o agendamento, controle e acesso remoto aos recursos de um cenário de redes real.

2.1.2 Objetivos Específicos

- Realizar estudo sobre equipamentos e protocolos de redes que serão utilizados no cenário da implantação;
- Analisar os requisitos necessários para a realização do desenvolvimento da aplicação;
- Configurar um servidor para permitir acesso web aos equipamentos de rede via comunicação serial;
- Configurar um cenário físico de equipamentos que serão utilizados na virtualização do cenário;
- Desenvolver um ambiente de aprendizagem web para o cadastramento de alunos, professores e equipamentos, assim como controlar o agendamento dos equipamentos e, por fim, desenvolver uma interface que permita os usuários configurar e manipular os equipamentos remotamente;

2.2 JUSTIFICATIVA

O sistema de educação vem mudando sua forma de disponibilizar conteúdos e propagar o conhecimento, este fato se dá devido a imersão de todas as áreas da sociedade na tecnologia (FERREIRA, DE LIMA, et al., 2013).

O novo formato de ensino exige cada vez mais, a ligação do ensino tradicional em sala de aula com formas atuais e tecnológicas. Essas formas visam cada vez mais deixar todos os conteúdos e recursos físicos e lógicos disponíveis para todos, em qualquer lugar e em qualquer hora.

Nas disciplinas de redes de computadores, que envolvem o uso de equipamentos de redes, que são caros e, por este fato, muitas vezes limitados, proporcionam uma experiência importante para o futuro do acadêmico. Os equipamentos possuem acesso controlado aos acadêmicos e professores, e por ter seu acesso restritos aos horários de aulas, tornam a realização de atividades e exercícios de fixação, extremamente dependentes de ferramentas de simulação, que não proporcionam, para o acadêmico, a mesma experiência de uma rede real,.

O projeto em questão visa minimizar o problema realizando a construção de um laboratório de redes virtual em uma plataforma web que vai realizar o gerenciamento e a disponibilidade de recursos físicos e lógicos, com a finalidade de gestão para os docentes e discentes. Estes recursos poderão ser gerenciados, liberados e utilizados com um controle de acesso.

2.3 ESTRUTURA DO TRABALHO

Esta monografia está organizado em capítulos, dos quais este é o primeiro e apresenta a ideia e o contexto do trabalho, incluindo os objetivos e a justificativa.

O Capítulo 2 contém o referencial teórico. O referencial teórico apresenta os conceitos envolvendo definições de redes de computadores, de estruturas e de programação.

No Capítulo 3 estão os materiais e a metodologia empregados no desenvolvimento deste trabalho.

O Capítulo 4 contém o resultado do desenvolvimento deste trabalho, como por exemplo, telas da aplicação, códigos essenciais para o desenvolvimento, entre outros.

O Capítulo 5 apresenta a conclusão com as considerações finais.

3 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os estudos e embasamentos teóricos que são utilizados no desenvolvimento deste trabalho.

3.1 ORGANIZAÇÃO E FUNCIONAMENTO DE REDE DE COMPUTADORES

Segundo Evandro Cantú (2003), a definição de rede de computadores é quando existe a conexão e a troca de dados entre dois ou mais dispositivos, entretanto, essa conexão acontece em diferentes escalas. A forma mais simples de rede consiste na ligação de dois dispositivos ligados fisicamente, denominada de conexão ponto a ponto. Essa ligação é ilustrada na Figura 1. O meio físico que busca realizar a ligação entre os dispositivos de rede tem como nome enlace de comunicação, podendo ser realizada por diversos tipos de conectores.



Figura 1 - Rede ponto a ponto.
Fonte: Autoria Própria.

Quando uma rede é composta pela conexão com vários dispositivos trata-se de um enlace multiponto (Figura 2), Este modelo possui vários tipos de dispositivos interligados com diferentes tipos de topologias.

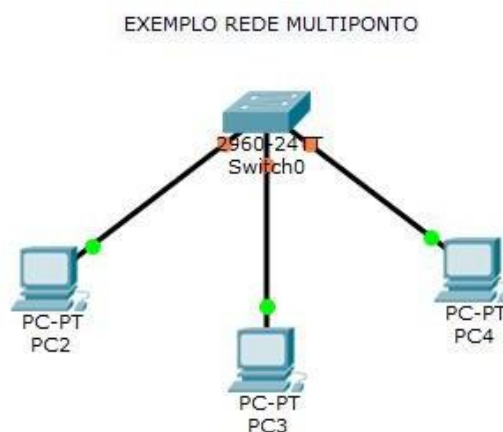


Figura 2 - Rede Multiponto
Fonte: Autoria Própria.

Uma topologia de rede tem como objetivo descrever como a rede de computadores é estruturada. Essa definição pode ser em relação a parte física, ou seja, demonstra com os computadores estão dispersos na rede ou em relação a parte lógica que demonstra uma relação de como os dados trafegam na rede (FRANCISCATTO, CRISTO, PERLIN, 2014).

De acordo com Tanenbaum (2003), redes de computadores trabalham enviando dados de um ponto a outro na rede por meio de diversos caminhos, em que é feita a divisão dos pacotes que irão ser roteados de forma independente. Os algoritmos de roteamento são responsáveis pela análise dos pacotes e definição de quais os melhores caminhos para cada determinado destino, isto é, identificar a melhor rota para os pacotes chegarem até o destino. Felipetti (2011) define roteamento como um conjunto de regras, que tem como objetivo trilhar o caminho que os dados originados em uma determinada sub-rede deverão percorrer para alcançar uma outra sub-rede. Cada algoritmo de roteamento tem uma métrica de distância administrativa diferente que atende suas características (FRANCISCATTO, CRISTO, PERLIN, 2014).

A Internet, com isso, nada mais é que um conjunto de redes conectadas entre si, onde existem roteadores que fazem toda a organização da rede por meio de seus protocolos de roteamento. Essa comunicação na Internet tem como base o protocolo IP (*Internet Protocol*),

o qual é responsável por endereçar e encaminhar os pacotes. É ele que permite que os dispositivos se conectem e sejam identificados na Internet (FELIPETTI, 2011).

3.2 PROTOCOLO IP

O protocolo IP trata-se de um dos protocolos mais importantes para o funcionamento da Internet, Guerber (2007) cita sua importância devido ao fato que, neste protocolo são armazenadas informações do endereço e as informações necessárias para se rotear um pacote em uma rede.

O endereço IP nada mais é que um endereço lógico que é responsável por determinar quem está enviando e quem receberá o pacote. Segundo Guerber (2007) todo o pacote trafegado, seja em uma rede local ou na Internet, conterá o endereço IP do remetente e o endereço IP do destino desse pacote.

3.3 MODELO OSI

O modelo OSI (*Open Systems Interconnections*) é um modelo de referência criado para a resolução de problemas em uma arquitetura de rede. Este modelo se baseia em uma proposta desenvolvida em 1984 pela organização ISO (*International Standards Organization*) (TANENBAUM, 2011).

O modelo é chamado de Modelo de Referência ISO/OSI (*Open Systems Interconnections*), este modelo trata da interconexão de sistemas abertos, ou seja, sistemas que estão abertos à comunicação e troca de dados com outros sistemas. Para atender toda a estrutura de comunicação, o modelo OSI é dividido em 7 camadas, sendo elas definidas por TANENBAUM:

Camada física: é a camada responsável por gerenciar a comunicação de mais baixo nível em uma arquitetura de rede computacional, nesta camada são transmitidos os bits por meio de

um canal de comunicação.

Camada de enlace de dados: Têm com sua principal função transformar um canal de transmissão de bits em uma comunicação em que a camada de rede consiga entender ou transformar um comunicação vinda da camada de rede para uma comunicação de bits.

Camada de rede: a responsabilidade desta camada é trabalhar com os datagramas e operações de transmissões em sub-redes. Nesta camada será determinado a maneira com que os pacotes serão roteados e encaminhados pelas redes de comunicação.

Camada de transporte: A função da camada de transporte é receber os dados das camadas, dividi-los em unidades menores, caso for necessário, e repassar as unidades para a camada de rede. Também será definido o tipo do serviço que será fornecido para as camadas relacionais e o tipo do transporte dos dados.

Camada de sessão: a camada de sessão tem como objetivo permitir que usuários de diferentes máquinas estabeleçam uma sessão entre eles.

Camada de apresentação: diferentemente das camadas mais baixas, a camada de apresentação precisa se preocupar em como as informações serão apresentadas, em relação, a sintaxe e a semântica das informações que são transmitidas.

Camada de aplicação: nesta camada são encontrados os protocolos que são necessários para o usuário realizar troca de dados e informações.

3.4 MODELO TCP/IP

Tanenbaum (2003) explica que o modelo TCP/IP (*Transmission Control Protocol / Internet Protocol*) foi criado com a intenção de suprir uma necessidade a qual o modelo OSI (*Open Systems Interconnections*) não atendia, que era de manter a rede ativa mesmo que ocorresse alguma falha, em suas sub-redes entre origem e destino.

Para resolver essas questões, alguns critérios na criação do protocolo TCP/IP foram estabelecidos, como um maior controle dos erros de transmissão de dados, criado o roteamento o qual encaminharia os pacotes de um roteador para outro e a utilização de endereços lógicos para identificar equipamentos na rede, nesse caso trata-se do protocolo IP (TANENBAUM, 2011).

Este modelo foi desenvolvido pelo Departamento de Defesa dos Estados Unidos da América, como preocupação de manter conexões estáveis enquanto as máquinas de origem e destino estivessem operantes, mesmo que por sua vez a conexão que intermediasse deixasse de operar.

KUROSE e ROSS citam que o modelo TCP/IP é dividido em quatro camadas, sendo elas:

Acesso à rede: é a primeira camada do modelo e é responsável pelo acesso à rede, realizando assim a transferência de dados entre os nós, dando assim suporte a camada de rede em que presta o serviço de acesso físico e lógico ao meio físico.

Rede: este nível é responsável por realizar a transferência de pacotes, ou datagramas, de um dispositivo qualquer para outro dispositivo, independentemente de sua localização na rede. É nessa camada que ocorre o roteamento dos pacotes, sendo feito o envio entre os roteadores pertencentes a rede entre destino e origem.

Transporte: responsável por prover um canal de comunicação lógico ponta a ponta entre os processos de aplicação que estão rodando em diferentes computadores, deixando assim de se preocupar com os detalhes da infraestrutura física usada para realizar o carregamento da mensagem.

Aplicação: é nessa camada que se encontram os protocolos que dão suporte às aplicações do usuário, como, por exemplo, para acessar páginas web é utilizado o protocolo HTTP (*Hypertext Transfer Protocol*). Para cada protocolo é utilizado uma porta, desse modo é possível saber qual o conteúdo do pacote de dados.

Na Figura 3 é possível visualizar a diferença do modelo OSI para o

modelo TCP/IP, definindo também a relação em ambos os modelos de referência.

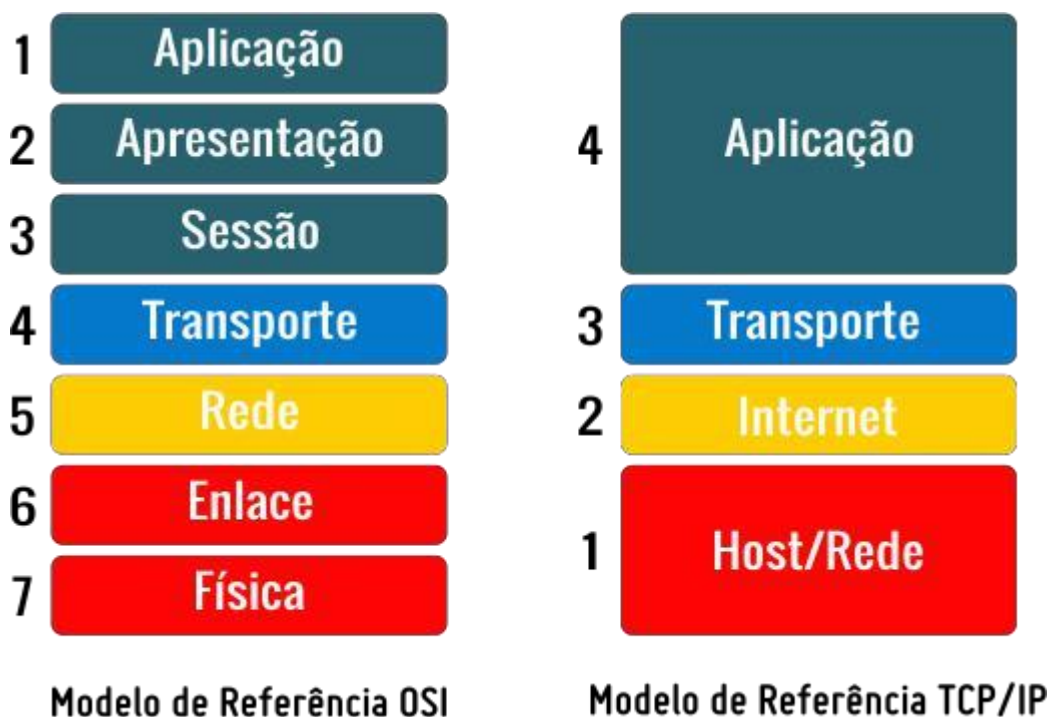


Figura 3 - Modelos de referência.

Fonte : <http://jkolb.com.br/modelo-osi-x-modelo-tcpip/>

3.5 EQUIPAMENTO DE REDE

Todos os equipamentos de rede de computadores utilizam e trabalham com protocolos de rede e, na sua maioria, se utilizam do modelo de referência TCP/IP para realizar a troca de informações.

Segundo Franciscatto, Cristo e Perlin, 2014, uma rede de computadores é composta por diferentes dispositivos, cada um com sua função, com o objetivo de dar funcionalidade e organização, bem como, prover a comunicação entre os diferentes componentes de uma rede.

Existem diversos modelos e diferentes tipos de equipamentos que compõem uma topologia de rede, eles variam de uma simples placa de rede a roteadores de alto desempenho.

3.5.1 Placa de rede

Dispositivos como computador e notebook, entre outros dispositivos, se conectam a uma rede de computadores por meio de uma placa de rede, que possibilita a conexão com a rede.

Placas de rede ou interfaces de rede, também denominadas de *Network Interface Card* (NIC), são equipamentos de comunicação inicial de um computador ou notebook, por exemplo, com os demais dispositivos da rede (switch, *hub*, roteadores, etc.), permitindo assim que estes dispositivos se conectem a uma determinada rede (FRANCISCATTO, CRISTO, PERLIN, 2014). Basicamente, a função de uma placa de rede é transmitir e receber dados da rede, tendo como principal função gerar sinais que serão captados na rede e controlar todo o fluxo de dados que entra e sai do computador. As placas de rede podem ser *on-board*, ou seja, uma placa que já vem integrada à placa-mãe, ou *off-board*, quando a placa é encaixada na placa-mãe do computador por meio dos slots de conexão, podendo ser visualizada na Figura 3 (FRANCISCATTO, CRISTO, PERLIN, 2014).



Figura 4 - Placa de rede.
Fonte: www.waz.com.br

3.5.2 Hub

Um *hub* é um dispositivo de rede que opera na camada física do modelo OSI, tem como objetivo centralizar os dados que trafegam pela rede, interligando assim vários dispositivos. Sua principal função em uma rede é receber o sinal de um dos computadores interligados a ele e

difundir este sinal para todos os outros dispositivos conectados, para que os dados possam ser recebidos pelo computador de destino (TANENBAUM, 2011).

O *hub* é composto de várias portas de conexão, mudando a quantidade de portas dependendo do modelo e do fabricante. As portas são responsáveis pela conexão de entrada via cabos de rede, permitindo desta forma que todos os computadores conectados a esse hub se comuniquem entre si (FRANCISCATTO, CRISTO, PERLIN, 2014).

A entrega de pacotes de um *hub* acontece de uma forma geral, ou seja, ao receber um pacote de um dispositivo qualquer o *hub* transmitirá esse pacote para todos os outros equipamentos buscando encontrar o destinatário deste pacote (FRANCISCATTO, CRISTO, PERLIN, 2014).

3.5.3 Switches

Classificado como substituto ao *hub*, o *switch* é um dispositivo de rede que tem o objetivo de interligar os computadores da rede e gerenciar a entrada, saída e a entrega de pacotes (TANENBAUM, 2011).

O *switch* diferentemente do *hub* recebe um pacote de um computador da rede e entrega ao computador destino, fazendo uma ligação entre emissor do pacote e receptor, por meio de uma porta de conexão. Já o *hub* conforme visto anteriormente, simplesmente recebe um pacote na rede e distribui a todos os micros da rede (FRANCISCATTO, CRISTO, PERLIN, 2014).

Através de um *switch* é possível chavear conexões entre os computadores que desejam se comunicar, permitindo que diversos computadores conversem entre si ao mesmo tempo, além de aumentar a taxa de transmissão da rede (FRANCISCATTO, CRISTO, PERLIN, 2014).

Quando o termo *switch* é usado deve-se tomar cuidado, os *Switches* são classificados de acordo com o nível que o dispositivo opera, podendo assim ser de camada 2 ou camada 3 do modelo OSI.

3.5.4 Roteadores

Roteadores são dispositivos que buscam direcionar pacotes com base em seus endereços lógicos (IP). Estes dispositivos geralmente interligam redes locais (*Local Area Network - LANS*) e redes geograficamente distribuídas (*Wide Area Network - WANS*) e trabalham utilizando uma tabela de roteamento (FOROUZAN, 2008).

Os roteadores têm como finalidade principal decidir o caminho dos pacotes de dados. Os roteadores operam na camada 3 (três) do modelo de referência ISO/OSI (*International Organization for Standardization / Open Systems Interconnection*).

Para realizar e estabelecer a rota, o roteador consulta uma tabela interna de roteamento que contém as informações sobre a rede, esta tabela pode ter seu preenchimento de forma estática ou dinâmica (FOROUZAN, 2008).

3.6 SIMULADORES E EMULADORES DE CENÁRIOS DE REDE.

Simuladores buscam por meio de softwares e hardwares simular ambientes e problemas do mundo real com a maior fidelidade possível. Um emulador por sua vez é uma ferramenta que reproduz uma plataforma virtualizada que permite que uma arquitetura de computador consiga executar sistemas que foram construídos em outras arquiteturas específicas (SOUZA, SILVA, et. Al, 2016).

SOUZA, SILVA, et. Al, (2016) definem que:

Os softwares emuladores, por sua vez, têm a capacidade de transformar um computador comum em um dispositivo de rede, como um roteador real, replicando praticamente todas as suas funções. A limitação fica por conta do desempenho do elemento emulado, notadamente inferior ao de um elemento físico. Por este motivo, é desaconselhável a aplicação de elementos emulados em testes de desempenho.

Esses equipamentos buscam reproduzir o comportamento de um sistema sob determinadas condições de adversidade, o que permite aos usuários que trabalham com estes softwares praticar em um ambiente de perigo não real. A Figura 4 mostra um simulador de voo.



Figura 5 - Simulador de voo.

Fonte : <https://airway.uol.com.br/fab-aciona-novo-simulador-de-voo-do-c-105-amazonas/>

Flato e Guimarães (2011) adicionam ainda que: a simulação é um aprendizado com situações “próximas do real”, que se correlaciona com retenção do conhecimento por um tempo mais prolongado e absorção do conteúdo de forma mais agradável.

Os simuladores e emuladores estão empregados em todas as áreas possibilitando, assim, o treinamento e a busca do aperfeiçoamento, na educação isso não é diferente. Pôr meio da simulação é possível facilitar o aprendizado em qualquer área, minimizando assim, as possíveis complicações (SOUZA, SILVA, et. Al, 2016).

Simuladores e emuladores tem um papel fundamental no auxílio do aprendizado do funcionamento de redes de computadores. Eles servem para a criação de cenários, buscando aproximar o modo de operação de um ambiente real antes da implantação dos mesmos, neste cenário os dois principais simuladores são GNS3 e Packet Tracer.

3.6.1 GNS3

O Graphical Network Simulator (GNS3) é uma ferramenta para a simulação de redes de computadores que permite a criação de redes por

meio da emulação de diversos tipos e modelos de equipamentos. O GNS3 busca simular cada dispositivo com o seu sistema de operacional real, tornando o seu uso o mais próximo do ambiente real. Segundo FILIPPETTI (2011), o GNS3 suporta uma gama enorme de funcionalidades que o tornam, definitivamente, um dos melhores simuladores de redes do mercado.

A ferramenta GNS3 nasceu com o objetivo de auxiliar os estudantes e profissionais da comunidade em geral que tem como objetivo alavancar uma certificação na área de redes ou simplesmente para simular redes com equipamentos diversos, já que o custo de aquisição de equipamentos para fins educacionais seriam relativamente inviável (KNEBEL, BATTISTI, 2012).

Com a evolução de equipamentos e tecnologias e com o passar do tempo outros recursos foram adicionados ao GNS3, ampliando assim a possibilidade de simular outros equipamentos de forma real.

Para a emulação dos equipamentos o GNS3 necessita da aquisição das imagens ou sistemas operacionais de cada equipamento. A Figura 6 representa uma tela do GNS3, demonstrando a lista de equipamentos disponível assim como o cenário de rede;

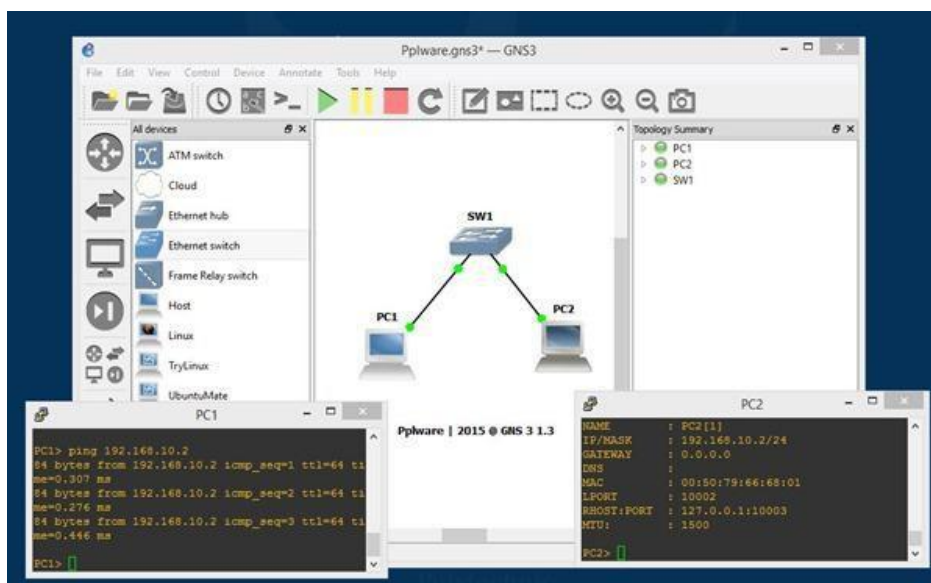


Figura 6 - Tela do GNS3
Fonte: Autoria Própria.

Outro recurso interessante do GNS3 é a capacidade dessa

ferramenta de se comunicar com o mundo real por meio do mapeamento de portas, possibilitando conectar equipamentos reais aos cenários simulados na ferramenta (KNEBEL, BATTISTI, 2012).

3.6.2 Cisco Packet Tracer

Outro simulador usado para o ambiente de redes é o Cisco Packet Tracer. Este simulador é desenvolvido pela Cisco Systems para ser utilizado como ferramenta na Academia Cisco (*Cisco Networking Academy*) um programa mundial de treinamento de novos profissionais na área de redes de computadores (SOUZA, SILVA, et. Al, 2016).

Por meio desse simulador é possível criar topologias com variadas infraestruturas de redes, seja ela, simples ou complexas. A Figura 7 demonstra a interface do Cisco Packet Tracer com uma topologia rede.

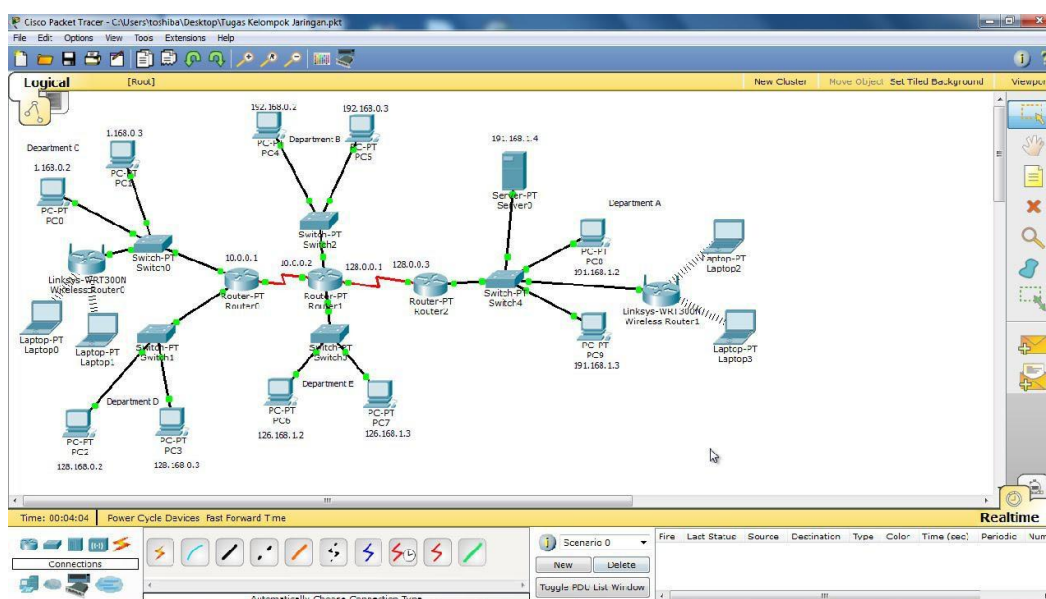


Figura 7 - Tela do Packet Tracer
Fonte: Autoria Própria.

3.6.3 NETLAB CISCO

Os laboratórios remotos estão se tornando um ambiente educacional muito comum e seu uso cresce em todo mundo. O Laboratório remoto NETLAB mostrado na Figura 8 tem como principal objetivo a

realização de uma interação com equipamentos de rede reais. Essa interação busca ser realizada com a criação de um cenário virtual (NETLAB+, 2018).

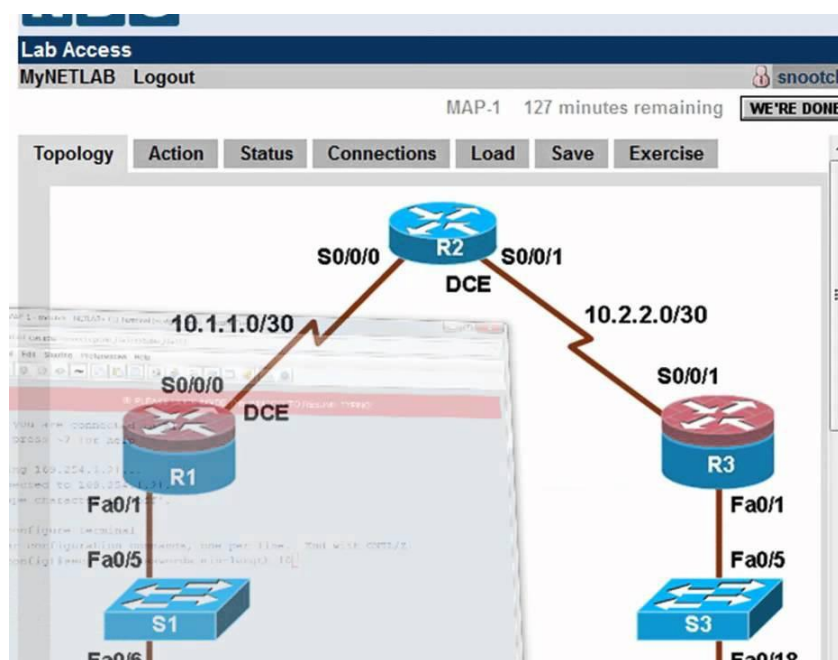


Figura 8 - Tela do NETLAB CISCO.
Fonte: <https://goo.gl/reS1dE>

O NETLAB é uma ferramenta da Cisco que permite aos usuários o cadastramento, configuração e manuseio de equipamentos de redes de uma forma virtual, o grande problema da ferramenta NETLAB, por ser uma ferramenta proprietária é seu custo de aquisição que pode variar de \$ 9,995 USD até \$ 39,980 USD para sua aquisição além de uma licença anual que pode variar de \$ 1,995 USD até \$ 5,995 USD (NETLAB+, 2018).

3.7 ENGENHARIA DE SOFTWARE.

Os softwares estão se tornando cada vez mais incorporados em nosso cotidiano, cada vez mais complexos e tendo um elevado número de pessoas relacionadas a ele. Para isso é necessário que todas as funcionalidades sejam levantadas e estudadas para atender essas pessoas que vão utilizar os softwares (PRESSMAN, 2011).

Quando uma aplicação está para ser desenvolvida, o número de

peças envolvidas no projeto vai influenciar diretamente no tamanho do software, a complexidade, o número de ideias e funcionalidades pois cada uma das pessoas envolvidas no projeto pode ter visões diferentes de como o software deve funcionar. Nesse momento a engenharia de software entra em ação (PRESSMAN, 2011).

A engenharia de software é essencial para o sucesso de um projeto, está relacionada diretamente com a busca na qualidade de um software visando atender os requisitos levantando por ela (PRESSMAN, 2011).

PRESSMAN (2011, p39) fala que “software, em todas as suas formas e em todos os seus campos de aplicação, deve passar pelos processos de engenharia.”

A engenharia de software é uma tecnologia em camadas. Qualquer abordagem de engenharia (inclusive engenharia de software) deve estar fundamentada em comprometimento organizacional com a qualidade [...] leva ao desenvolvimento de abordagens cada vez mais efetivas na engenharia de software. A pedra fundamental que sustenta a engenharia de software é o foco na qualidade. PRESSMAN (2011, p39).

3.8 ENGENHARIA DE REQUISITOS

Requisitos são os pontos principais na análise de um software, nesse ponto são identificados os pontos indispensáveis e necessários para a realização do projeto. Esses requisitos levantados serão a base que levará o projeto a ser desenvolvido, eles serão responsáveis pelo fracasso ou sucesso do projeto (PIRES, 2012).

QUITEIRO A. P, apud (IEEE 1990) a análise de requisitos é um processo que envolve o estudo das necessidades do usuário para se encontrar uma definição correta ou completa do sistema ou requisito de software.

O processo para construir um software deve ser iniciado pelo levantamento das funcionalidades que conterá essas funcionalidades são chamados de requisitos, os requisitos são necessários para entender as regras do negócio (PRESSMAN, 2011).

O amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos e denominado engenharia de requisitos. Na perspectiva do processo de software, a engenharia de requisitos é uma ação de engenharia de software importante que se inicia durante a atividade de comunicação e continua na de modelagem. Ela deve ser adaptada as necessidades do processo do projeto, do produto e das pessoas que estão realizando o trabalho PRESSMAN (2011, p127).

O processo para levantar os requisitos deve ser realizado com interessados no projeto, ou seja, pessoas que estão diretamente envolvidas no projeto, esse levantamento e a etapa inicial para o desenvolvimento do projeto segundo PRESSMAN (2011, p127) “a engenharia de requisitos constrói uma ponte para o projeto e para a construção;”, e nesse processo que será delimitado o tamanho do software, ou seja, será definido o que conterà no software.

A engenharia de requisitos fornece o mecanismo apropriado para entender aquilo que o cliente deseja, analisando as necessidades, avaliando a viabilidade, negociando uma solução razoável, especificando a solução sem ambiguidades, validando a especificação e gerenciando as necessidades PRESSMAN (2011, p127).

O processo de levantamento dos requisitos aparentemente parece fácil, entretanto, existe alguns problemas que aparecem junto como esse processo, problemas com o limite do software podendo assim prejudicar o desenvolvimento devido a informações desnecessárias, os clientes não estão certos do que desejam.

Os problemas mais graves encontrados na área de engenharia de software são as mudanças de requisitos que ocorre frequentemente nas regras de negócios, como isso, deve-se escolher adequadamente as pessoas que vão participar do levantamento de requisitos, deixando sempre um espaço para mudanças nos requisitos levantados PRESSMAN (2011, p127).

3.9 ANÁLISE ORIENTADA A OBJETOS (UML)

A análise é o primeiro passo a ser dado na construção de um projeto, nela será definido todos os pontos que o projeto deverá ter, atender e também deve conter todas as regras que deve seguir.

Esta análise é realizada com as informações obtidas junto com o cliente ou em cima de uma análise de um software já desenvolvido. Essas informações serão transformadas em requisitos no processo de levantamento (PRESSMAN, 2011).

O *Unified Modeling Language (UML)* é uma linguagem de modelagem que permite estruturar projetos de software. É importante visar que o UML não é uma metodologia, ou seja, ela não indica uma sequência de passos que deve ser seguida para se chegar ao objetivo que é a diagramação (SEABRA, 2001). UML permite modular um software de forma que o usuário possa ter uma nova perspectiva do software, isso é possível por meio da criação de diagramas estruturais e diagramas comportamentais (PRESSMAN, 2011).

3.9.1 Diagramas estruturais

Diagramas estruturais são diagramas que representam a estrutura de um sistema. Eles são classificados em diagramas de classe, objeto, de pacotes, de estrutura composta, de componentes e de implantação (PRESSMAN, 2011).

O diagrama de classe é o diagrama que mais chega perto do código do programa, nesse diagrama mostrado na Figura 9 serão indicadas as classes, métodos, atributos e os relacionamentos todos envolvidos.

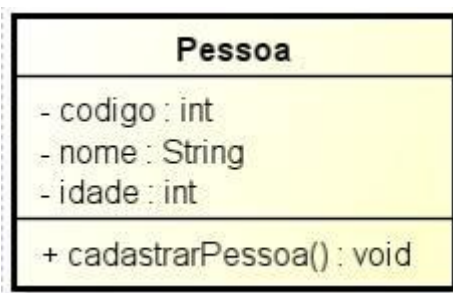


Figura 9 - Diagrama de classe estrutural.
Fonte: Autoria Própria.

3.9.2 Diagramas comportamentais.

Diagramas comportamentais são utilizados para visualizar, especificar, construir e documentar os aspectos dinâmicos de um software. Eles são classificados em diagrama de caso de uso, sequência, colaboração, estados e atividade, nesse trabalho foi utilizado apenas diagrama de caso de uso (PRESSMAN, 2011).

Os Diagramas de caso de uso mostrado na Figura 10 definem uma sequência de ações que o sistema realizará afim de gerar para o usuário uma visão de como isso funcionará.

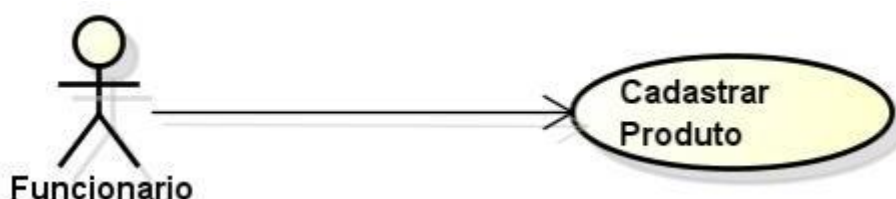


Figura 10 - Diagrama de caso de uso.
Fonte: Autoria Própria.

3.10 DESIGN DE INTERAÇÃO.

Design de interação combina estudos, pesquisas e habilidades para a identificação das necessidades dos usuários e com isso produzir soluções que visão melhorar a interação do usuário com o software.

A principal preocupação do termo design de interação é desenvolver produtos que sejam utilizáveis, ou seja, que os produtos desenvolvidos sejam de fácil aprendizagem, que sejam eficazes no uso, que proporcionem uma experiência agradável para o usuário, isso é atingido por meio de técnicas e métodos (BRAGA, 2007).

O design de interação também é realizado utilizando as necessidades dos usuários, tratando de pegar as necessidades e tornando-as eficientes, para que sejam altamente produtivas para o trabalho do usuário.

3.10.1 Prototipação

A prototipação é uma técnica para representar as telas e processos antes do desenvolvimento. Esse passo simula a usabilidade do software junto com os envolvidos no projeto (BARBOSA, SILVA, 2010).

A prototipação pode ser realizada por meio de manuais como desenho em um papel até a utilização de software para fazer a prototipagem. De acordo com Barbosa, Silva (2010, p. 358): “Simular o uso em papel é um modo rápido e barato de identificar problemas de usabilidade antes mesmo de construir uma solução em IHC executável.”

Braga argumenta sobre prototipagem de interfaces:

A prototipação de interfaces aparece como uma técnica que pode ser utilizada com sucesso como valioso auxiliar na determinação dos requisitos do problema. De modo geral, os atores ou usuários interagem com o software via interfaces, e elas são a sua metáfora de trabalho. Nas fases iniciais, em que os requisitos ainda são nebulosos, a prototipação de interfaces pode ser um fator decisivo no sentido de tornar o processo mais rápido e seguro. Interagindo com os protótipos de interfaces, os usuários podem ser compelidos a ir falando sobre a sua interação com elas, se estão faltando funcionalidades, se os passos para resolver problemas estão corretos, etc. BRAGA (2007).

3.10.2 Design de interface

O design de interface faz a mediação entre interação dos usuários e o sistema de informação, essa interação é chamada de interface do usuário (IU), consiste em ser tudo que as pessoas entram em contato com

o sistema, seja, físico, perceptiva ou conceitualmente (BENYON, 2011).

Fisicamente as pessoas interagem com o sistema de diversas formas, um simples clique em um botão, toques em telas isso e muito mais acarreta em um contato físico entre homem e máquina.

Perceptivamente podemos dizer que são as interações com o sistema por meio da visão, audição e toque. Os designs devem ser elaborados de forma que aos primeiros momentos com o software exista uma percepção imediata de coisas na tela (BENYON, 2011). Conceitualmente, é a forma de interação a qual as pessoas usam os sistemas e os dispositivos por meio do conhecimento, de como fazer e o que eles podem fazer.

BENYON (2011, p 213): “O design de interface preocupa-se em criar uma experiência que permite as pessoas utilizarem, da melhor forma possível o sistema que é objeto do design.”.

4 MATERIAIS E METODOLOGIA

A ênfase deste capítulo está em reportar o que será utilizado e as configurações das tecnologias que são necessárias para implementar o aplicativo e/ou executá-lo, Para a mostrar quais as versões utilizadas no desenvolvimento do software foi criado a Tabela 1.

Material	Versões
Python	3.7.1
Flask	1.0.2
NodeJS	10.14.2
Angular	6.1.10
Angular CLI	6.2.8
PostgreSQL	11.1
Astah	Community

Tabela 1 - Tabela de versões.
Fonte : Autoria própria.

4.1 MATERIAIS

4.1.1 Python

Python é uma linguagem de programação criada no fim dos anos 80 por Guido Van Rossum, permite por meio de uma forma fácil e rápida desenvolver e integrar sistemas *web* ou *desktop*.

Para Dubois, 2007:

A linguagem de programação Python está se estabelecendo como uma das linguagens mais populares para computação científica. Graças à sua natureza interativa de alto nível e ao seu amadurecimento do ecossistema de bibliotecas, é uma opção atraente para desenvolvimento algorítmico e análise exploratória de dados.

Python possui em seu propósito geral o desenvolvimento de aplicações, não se limitando a nenhum segmento, com isso, o trabalhando em conjunto com alguns *frameworks* vem se tornando uma linguagem

bastante produtiva em relação ao desenvolvimento (Dubois, 2007).

4.1.2 Flask

Flask é um framework Web desenvolvido para a utilização com Python de alto nível. Busca assim como o python ter um desenvolvimento rápido e o *design* limpo e pragmático. (FLASK, 2018).

4.1.3 PostgreSQL

PostgreSQL é um banco de dados livre criado em 1995, consiste em ser um sistema de gerenciamento de banco de dados objeto relacional utilizado em vários segmentos (POSTGRESQL, 2018).

4.1.4 Node.js

NodeJS é um *runtime* assíncrono orientado a eventos que trabalha com JavaScript no lado do servidor, tem como proposta oferecer serviços e bibliotecas padrões para os desenvolvedores web (CANTELON, et al, 2014).

Desde que o Node.js apareceu em 2009, o JavaScript passou de uma forma pouco tolerada para um dos idiomas mais importantes para todos os tipos de desenvolvimento de software. Isso se deve em parte à chegada do ECMAScript 2015, que resolveu vários problemas críticos em versões anteriores da linguagem (CANTELON, et al, 2014).

O Node.js tem como característica rodar em uma única thread. Isso se caracteriza como um problema em várias linguagens, porém o Node.js segue o modelo usado nos navegadores, onde escrevem-se sequências de instruções que são executadas uma de cada vez (CANTELON, et al, 2014).

4.1.5 HTML

O Hypertext Markup Language (HTML) é uma linguagem de marcação de Hipertexto. Esta linguagem é entendida universalmente por diversos meios de acessos Computadores, *Smartphones*, *Smart Tv*, etc.

HTML 5 é o novo padrão html, criado para substituir o HTML 4 que foi desenvolvido em 1999. Com a chegada do HTML o termo internet mudou significativamente.

Ferreira e Eis, argumentam sobre o HTML5:

Um dos principais objetivos do HTML5 é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final.

O HTML 5 é uma junção de CSS3 (*Cascading Style Sheets*), Java Script e Html, com isso permite por meio de suas APIs a manipulação desses elemento fazendo assim com que as aplicações se tornarem leves e funcionais.

4.1.6 Web Services

Um *Web Service* é um conjunto de métodos desenvolvidos em uma determinada linguagem de programação que poderá ser acessada via protocolo HTTP, ou seja, significa que aplicações desenvolvidas em uma linguagem poderão acessar aplicações em uma linguagem diferentes utilizando como a via http.

Para Haddad, 2013:

O uso de Web Service simplesmente é fantástico porque você pode acessar rotinas de validação de cartão de crédito, endereçamento postal (CEP), calcular valores de fretes dos sites de comércio eletrônico, news de empresas, enfim, uma gama gigantesca de produtos. A grande diferença em relação ao desenvolvimento atual é que a empresa que desenvolve uma DLL, não precisa ficar distribuindo para todos os clientes, esta estará armazenada em um único lugar que

será acessada via http. Isso sem falar em atualizações de DLL atuais, o que pode causar um transtorno para todos.

Com a utilização de Web Services também é possível interagir novas aplicações com aplicações antigas e que foram desenvolvidos em plataformas diferentes sejam totalmente compatíveis isso e feito pelas aplicações por meio de envio e chegada de dados do formato *Extensible Markup Language (XML)*, as aplicações transformam esse XML que é uma linguagem universal para sua própria linguagem (HADDAD, 2013)

4.1.7 Angular

O Angular é uma plataforma escrita em TypeScript que facilita a criação de aplicativos com a web e mobile, combinando modelos declarativos, injeção de dependência, ferramentas de ponta a ponta e práticas, que são recomendadas para resolver desafios do desenvolvimento.

TypeScript por sua vez é um conjunto de bibliotecas escrito em JavaScript que foi criado pela Microsoft permitindo a criação de scripts com a utilização de tipagem estática, orientação a objeto entre outras facilidades.

Angular permite que os desenvolvedores criem aplicativos na *Web*, em dispositivos móveis ou na área de trabalho, por meio de uma plataforma estruturada que permite a criação desses aplicativos em HTML e TypeScript.

A construção de uma aplicação em Angular está toda baseada em módulos, componentes e serviços. Os módulos fornecem um contexto de compilação para os componentes, basicamente os módulos chamados no angular de NgModules coletam códigos relacionados as funcionalidades e aplicam isso para a execução do aplicativo.

Os componentes em Angular são de extrema importância, tudo em Angular é um potencial componente. Com isso, cada aplicação tem, pelo menos, um componente, este que é chamado de componente raiz.

Componentes, módulos e serviços são combinados em uma

aplicação Angular, conforme mostrado na Figura 11 para criar uma aplicação totalmente estruturada. Cada componente define uma classe que contém dados e lógica do aplicativo e está associada a um modelo HTML que define uma exibição de um ambiente.

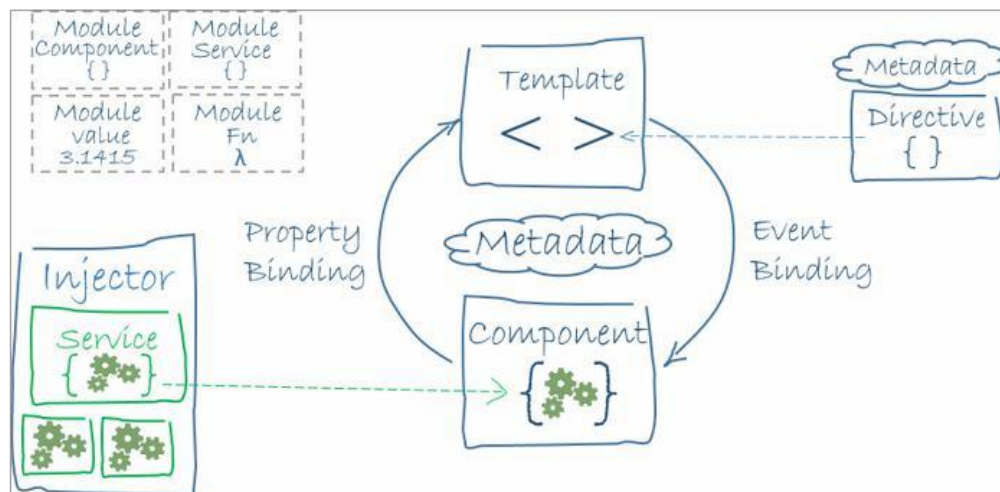


Figura 11 - Estrutura básica do angular.
Fonte: <https://angular.io/guide/architecture>

4.1.7.1 AngularCLI

A criação de um projeto angular não é exatamente uma tarefa simples, pois existem diversas bibliotecas, *frameworks* e configurações que precisam ser importadas e inicializada no projeto. Para resolver este problema foi criado a ferramenta denominada AngularCLI.

O AngularCLI é uma ferramenta para iniciar, desenvolver, sustentar e manter aplicações desenvolvidas em angular, com a utilização desta ferramenta o processo de criação de aplicativos em angular se torna mais ágil e produtiva.

Para a criação e utilização do AngularCLI em um projeto, é necessário ter instalado e já configurado o servidor de aplicação Nodejs apresentado anteriormente.

4.1.8 Astah

Astah é uma ferramenta computacional que auxilia na criação de

diagramas, utilizada na criação de diagramas estruturais e comportamentais, que são fundamentais nos processos de engenharia de softwares. Na Figura 12 é possível visualizar a interface de criação do Astah (MARTINS, 2014).

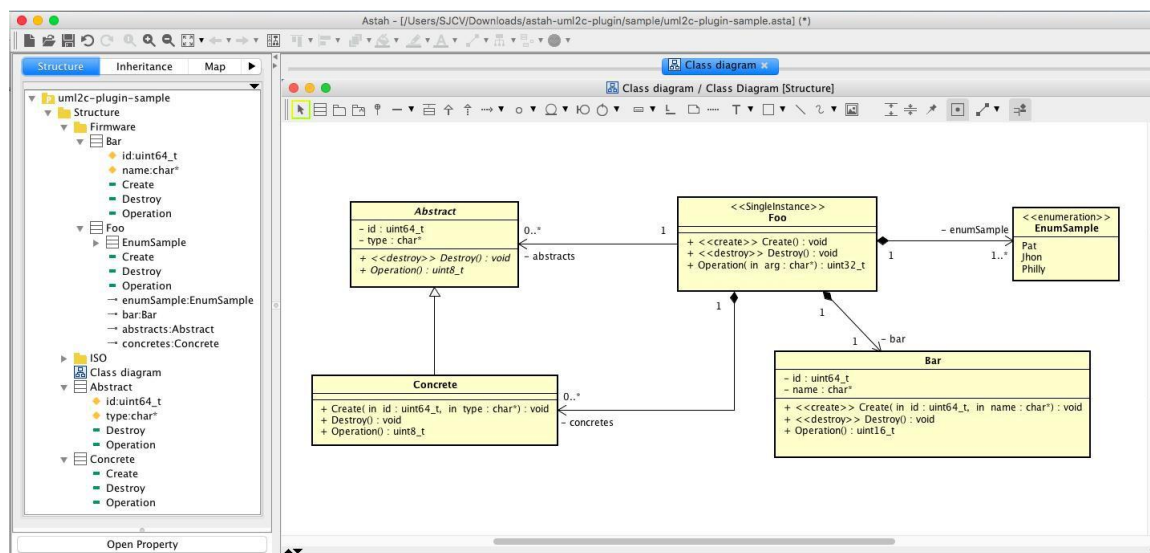


Figura 12 - Tela inicial do Astah.
Fonte: Autoria Própria.

Astah foi desenvolvido inicialmente por Kenji Hiranabe, em 1996. Na época, UML e alguns outros padrões de *design* começavam a aparecer no cenário mundial (MARTINS, 2014). Com a ferramenta Astah é possível criar diagramas de Classes, sequencias, caso de uso entre outros que estão disponíveis em versões comunitária e privadas (MARTINS, 2014).

4.1.9 Switch 2960 e Switch L3 3750

Para o trabalho em questão será utilizado para a criação de cenários de rede Switches do modelo Cisco 2960 mostrado na Figura 13 e 3750 mostrado na Figura 14, tendo a função de ligar os dispositivos da rede podendo assim ser manipulado pela plataforma em questão, ambos equipamentos utilizados são da marca Cisco.



Figura 13 - switch Cisco 2960
Fonte: cisco.com



Figura 14 - switch L3 3750
Fonte:cisco.com

4.1.10 Roteador 2811 e 3800

Roteadores tem como função definir e direcionar os pacotes em uma rede de computadores, para o projeto em questão os modelos Cisco 2811 demonstrado na Figura 15 e o modelo Cisco 3800 Series demonstrado na Figura 16, ambos equipamentos são da marca Cisco.



Figura 15 - Roteador Cisco 2811.
Fonte: Cisco.com



Figura 16 -Roteador Cisco 3845
Fonte: cisco.com

4.2 METODOLOGIA

O método utilizado para a realização do trabalho está baseado nas fases de análise de requisitos, prototipação, manipulação de hardware e criação de cenário de rede, desenvolvimento da plataforma e testes do ambiente, com isso foi utilizado o processo unificado o qual busca por meio de um a abordagem de interações incrementar o produto final, criando um fluxo de trabalho que é executado a cada ciclo de execução (KRUCHTEN, 2004).

No trabalho foi aplicado as atividades de: análise de Requisitos, análise e projeto, prototipação, desenvolvimento do software e testes aplicados na aplicação e no cenário de rede de computadores.

Análise de Requisitos: a definição dos requisitos foi

realizada a partir da descrição dos requisitos do ponto de vista dos professores e coordenadores do curso, além de utilizar o trabalho citado no referencial como base de conhecimento.

Análise do projeto: na análise do projeto os requisitos levantados foram utilizados e modelados sob a forma de diagrama de casos de uso e diagramas de classe.

Prototipação: com base no levantamento de requisitos e da análise e criação dos diagramas, será criado protótipos de telas utilizando técnicas de design.

Desenvolvimento: o desenvolvimento partiu dos levantamentos e prototipagem realizada anteriormente, este desenvolvimento busca atender as inclusões, exclusões, consultas e alterações. Buscando implementar as necessárias para o desenvolvimento da solução.

Testes aplicados: os testes foram aplicados de forma informal com o objetivo de identificar erro de codificação.

5 RESULTADOS

Nesta sessão será apresentado os resultados obtidos com os estudos e a análise do projeto em questão, buscando demonstrar todas as suas funcionalidades.

5.1 CRIAÇÃO DO CENÁRIO DE REDE

Para a criação do cenário de rede o usuário administrador terá todo o controle e flexibilidade para realizar a montagem do cenário ideal para seu problema, o pré-requisito para a criação e interligação do cenário com o servidor e realizar a comunicação via cabo serial, podendo utilizar ou não algum periférico de conexão. O próximo passo será realizar as configurações e cadastramentos que serão detalhados a seguir.

5.2 ANÁLISE DA PLATAFORMA VIRTUAL.

Para a construção do projeto foram realizadas conversas com alguns professores da área de redes, buscando levantar todos os requisitos necessários para o desenvolvimento do projeto.

O projeto em questão constitui-se na criação de uma aplicação web utilizando as ferramentas citadas acima, o principal objetivo é o controle, organização e reserva dos equipamentos.

O sistema deverá ter telas de cadastros, alterações e exclusões de alunos, turmas, equipamentos e reservas de equipamento. Esses cadastros deverão alimentar o modulo para gerenciar reservas e equipamentos.

Para representar as funcionalidades, organização e a abrangência do projeto foi elaborado um diagrama de caso de uso, mostrado na Figura 17 junto a um diagrama de classe, apresentado na Figura 18.

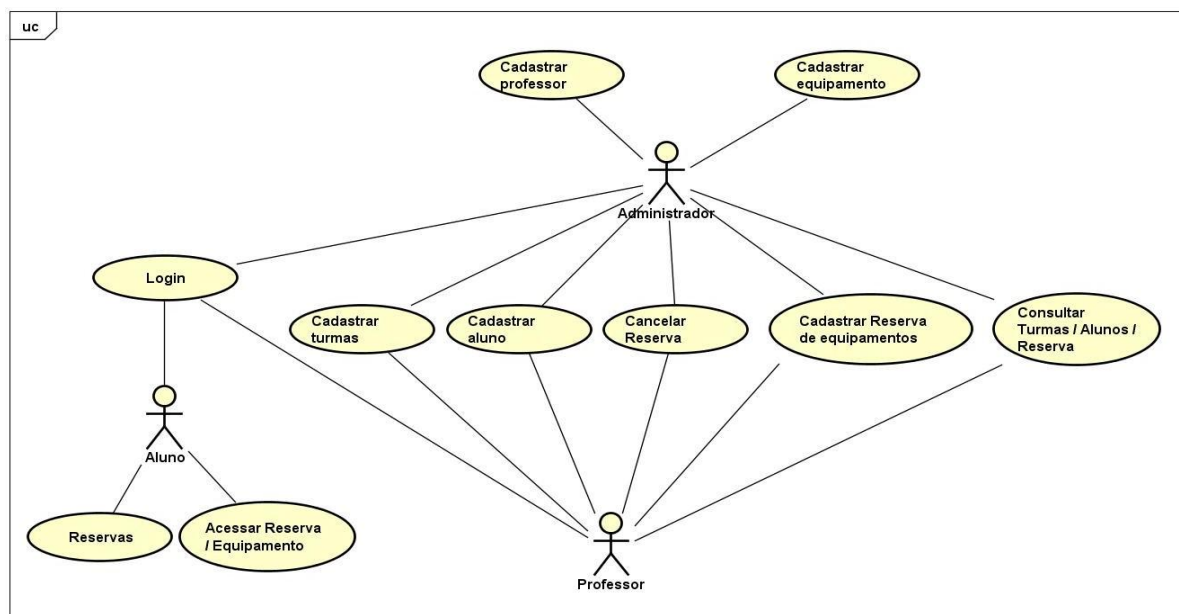


Figura 17 - Diagrama de caso de sistema
Fonte: Autoria Própria.

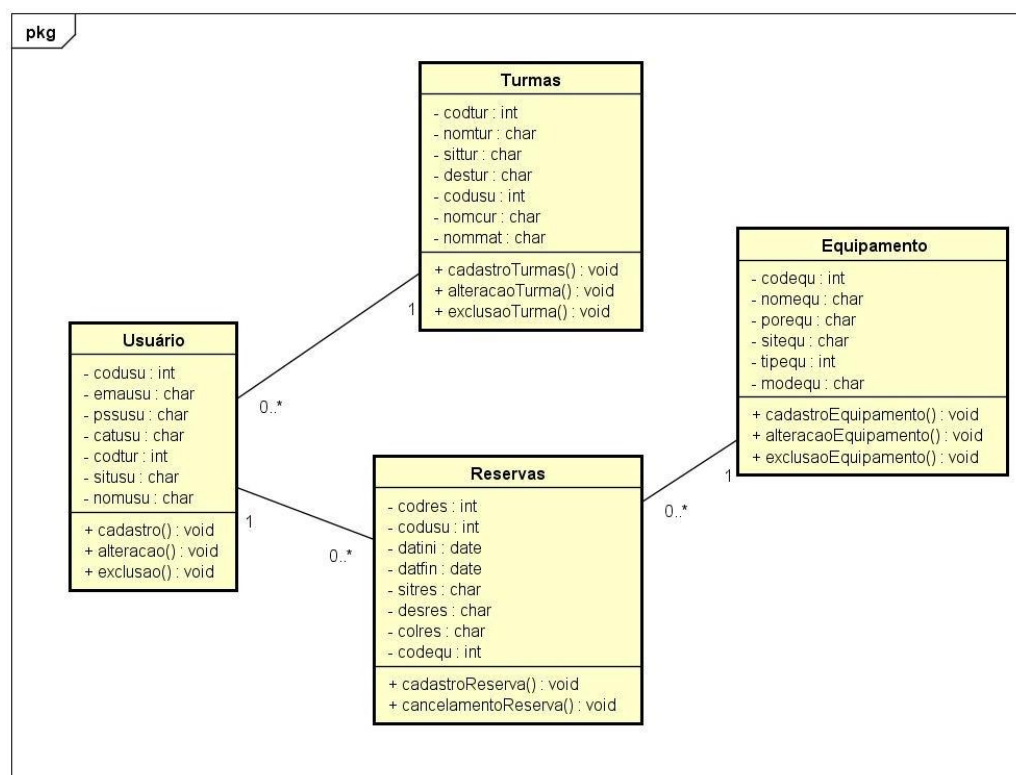


Figura 18 - Diagrama de classe do sistema
Fonte : Autoria Própria..

Para suprir os requisitos levantados e visualizados no diagrama de classe, algumas tabelas foram criadas sendo elas: usuários, turmas, equipamentos e reservas.

5.2.1 Usuários

Nos requisitos levantados foi necessário da criação de usuários, esta criação deve ser realizada com a definição de perfis e níveis de autorização diferentes dentro do sistema, sendo eles Administrador, Professor e alunos.

Aluno: o usuário do tipo aluno terá os privilégios de consultas de reservas e a utilização de equipamentos mediante a uma reserva realizada por um usuário Professor ou Administrador.

Professor: o usuário do tipo professor deve ser cadastrado por um outro usuário do tipo administrador, este usuário pode realizar as operações de : cadastro de alunos, cadastro de turmas, realização de reservas, cancelamento de reservas e visualização de logs de acesso a equipamentos.

Administrador: neste nível de autorização será possível realizar todas as operações de que um usuário do tipo professor pode realizar, além de realizar o cadastramento de um novo usuário administrativo ou professor e realizar o cadastramento, exclusão e alteração de equipamentos.

Para atender a demanda foi realizado a criação de uma base de dados com as características que são observadas na Quadro 1, que leva as informações do nome do campo, tipo do campo, qual o tamanho das informações que cabem no campo, descrição do campo e a informação se o campo é uma chave primária (PK) ou uma chave estrangeira (FK).

PK/FK	CAMPO	TIPO	TAMANHO	DESCRIÇÃO
PK	codusu	Integer		Código do usuário no sistema.
	email	String	150	E-mail do usuário.
	nome	String	150	Nome do usuário.

	pssusu	String	250	Senha do usuário.
	catusu	String	1	Categoria do usuário. A - Aluno. P - Professor M - Administrador
FK	codtur	Integer		Código da turma caso o usuário seja do tipo aluno.
	situsu	String	1	Situação do usuário. A - Ativo I - Inativo

Quadro 1 - Descrição de campos de usuários.

Fonte: Autoria Própria.

5.2.2 Turmas

Nos requisitos levantados foi necessário a criação de turmas para a identificação dos alunos e organização dos professores.

Para atender a demanda foi realizado a criação de uma base de dados com as características que são observadas no Quadro 2, que leva as informações do nome do campo, tipo do campo, qual o tamanho das informações que cabem no campo, descrição do campo e a informação se o campo é uma chave primária (PK) ou uma chave estrangeira (FK).

PK/FK	CAMPO	TIPO	TAMANHO	DESCRIÇÃO
PK	codtur	Integer		Código da turma no sistema.
	nomtur	String	150	Nome da turma.
	sittur	String	1	Situação da turma.
	destur	String	250	Descrição da turma.
	codusu	Integer	1	Código do responsável pela turma.
	nomcur	String	100	Nome do curso o qual será cadastrado alunos.
	nomma	String	100	Nome da matéria.

t			
---	--	--	--

Quadro 2 - Descrição de campos de turmas
Fonte: Autoria Própria.

5.2.3 Equipamentos

Nos requisitos levantados foi necessário a criação de uma forma de cadastrar e administrar quais os equipamentos serão utilizados para a criação do cenário.

Para atender a demanda foi realizado a criação de uma base de dados com as características que são observadas no Quadro 3, o qual leva as informações do nome do campo, tipo do campo, qual o tamanho das informações que cabem no campo, descrição do campo e a informação se o campo é uma chave primária (PK) ou uma chave estrangeira (FK).

PK/FK	CAMPO	TIPO	TAMANHO	DESCRIÇÃO
PK	codequ	Integer		Código do equipamento no sistema.
	nomequ u	String	150	Nome do equipamento.
	sitequ	String	1	Situação do equipamento.
	porequ	String	250	Porta utilizada para interligar o equipamento com o servidor.
	tipequ	String	1	Tipo do equipamento. R - Roteador. S - Switch.
	modequ u	String	100	Modelo do equipamento.

Quadro 3 - Descrição de campos dos equipamentos.
Fonte: Autoria Própria.

5.2.4 Reservas

Nos requisitos levantados foi necessário a criação de uma forma de

reservar os equipamentos que devidamente foram cadastrado assim como vincular esse equipamento a uma reserva.

Para atender a demanda foi realizado a criação de uma base de dados com as características que são observadas no Quadro 4, que leva as informações do nome do campo, tipo do campo, qual o tamanho das informações que cabem no campo, descrição do campo e a informação se o campo é uma chave primária (PK) ou uma chave estrangeira (FK).

PK/FK	CAMPO	TIPO	TAMANHO	DESCRIÇÃO
PK	Codres	Integer		Código da reserva no sistema.
FK	Codusu	Integer	150	Código do usuário vinculado a tabela e usuários.
FK	Codtur	Integer		Código da turma vinculado a tabela e usuários.
FK	Codequ	Integer		Código do equipamento.
	Datini	Datetime		Data e hora do início da reserva.
	Datfin	Datetime		Data e hora do final da reserva.
	Sitres	String	1	Situação da reserva
	Desres	String	200	Descrição da reserva.
	Colres	String	20	Cor do marcador da reserva

Quadro 4 - Descrição dos campos da reserva.

Fonte: Autoria própria

5.2.5 Prototipação das telas.

Para desenvolver os protótipos foram utilizados conhecimentos adquiridos por meio do estudo de interação com o usuário, visando tornar o sistema fácil e descomplicado para o usuário sem deixar de pensar no aspecto visual. Os protótipos foram desenvolvidos com a ferramenta Balsamiq.

Para os conceitos aplicados de menus e ícones o sistema busca ser

interativo, o agrupamento aplicado pelas área de informação que armazena, ou seja, cada opção terá dentro de si as funcionalidades relativas a sua área. Os ícones utilizados foram selecionados para ajudar o usuário a identificar a funcionalidade de seus respectivos botões, pode-se notar essa aplicação na Figura 19.

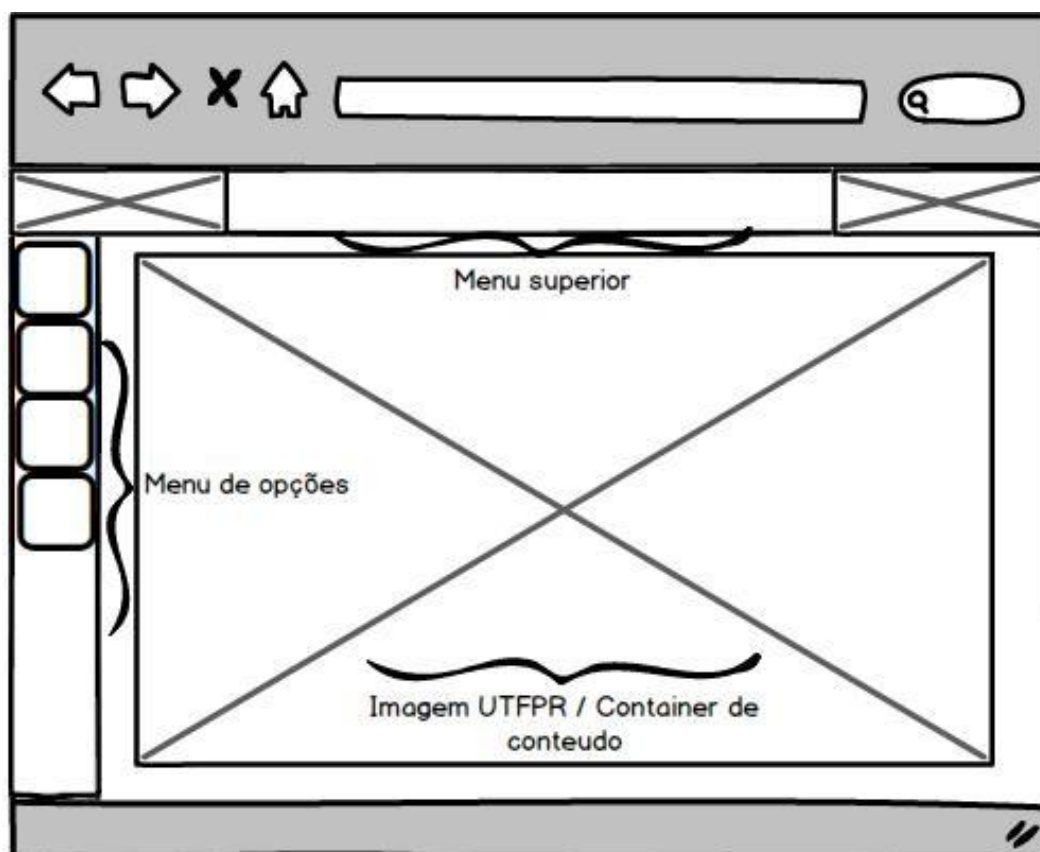


Figura 19 - Protótipo de tela principal
Fonte: Autoria Própria.

As telas de cadastro, alteração e exclusão por regra sempre será retornado para o usuário uma lista com todos os registros disponíveis, seja nas funcionalidades de usuários, turmas ou equipamentos. Os parâmetros que deverão ser preenchidos para suprir as necessidades do sistema estão descritos em tópicos anteriores. Na Figura 20 é possível visualizar qual será o padrão das telas de cadastro e consulta.

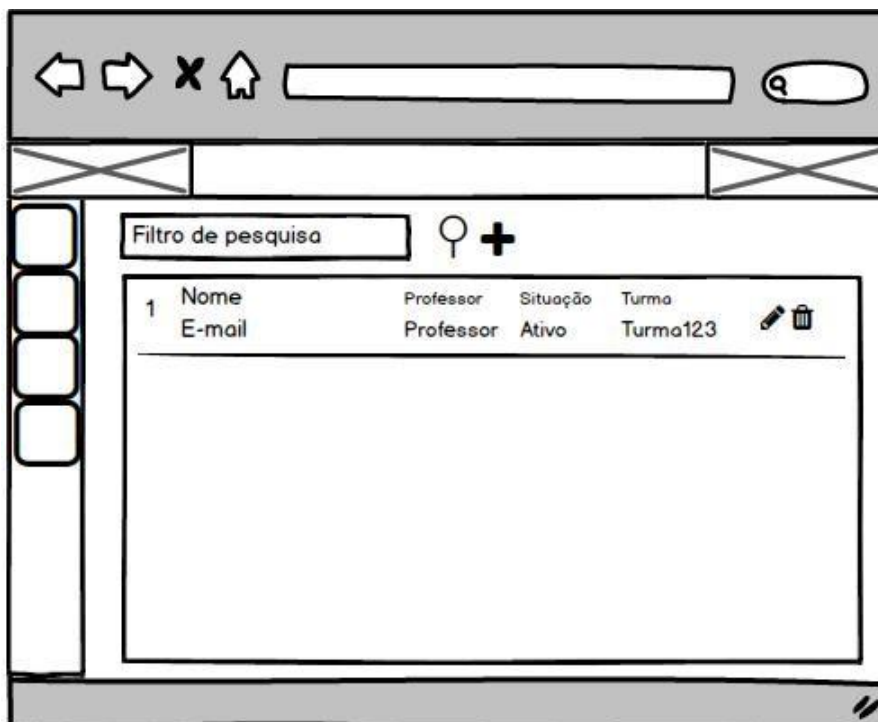


Figura 20 - Protótipo de telas de cadastro, consultas e exclusões
Fonte: Autoria Própria.

Para a realização dos cadastros ou alterações será aberto para o usuário uma tela de preenchimento de dados, mostrada na Figura 21.

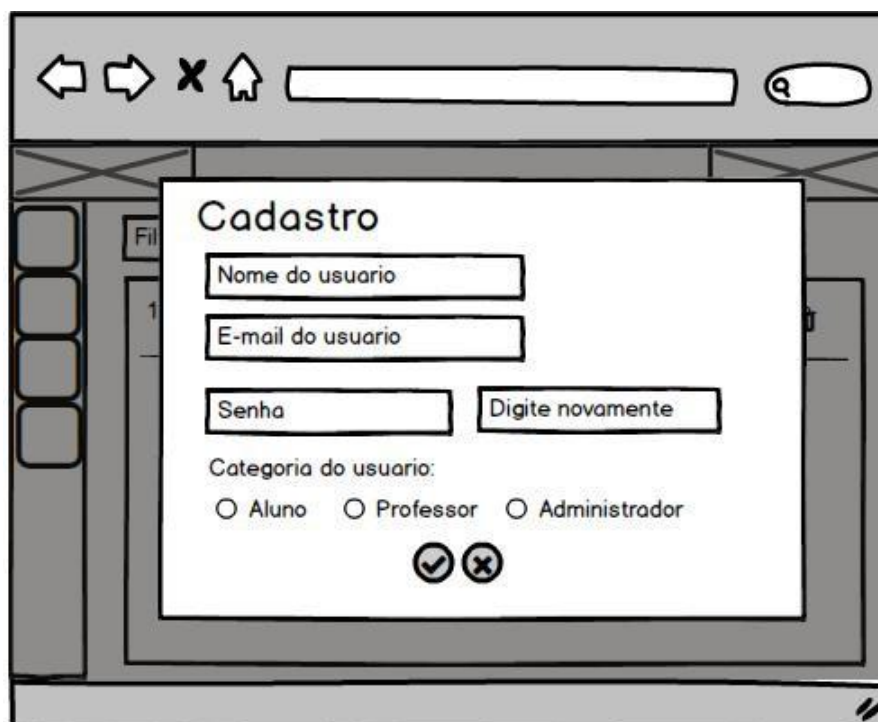
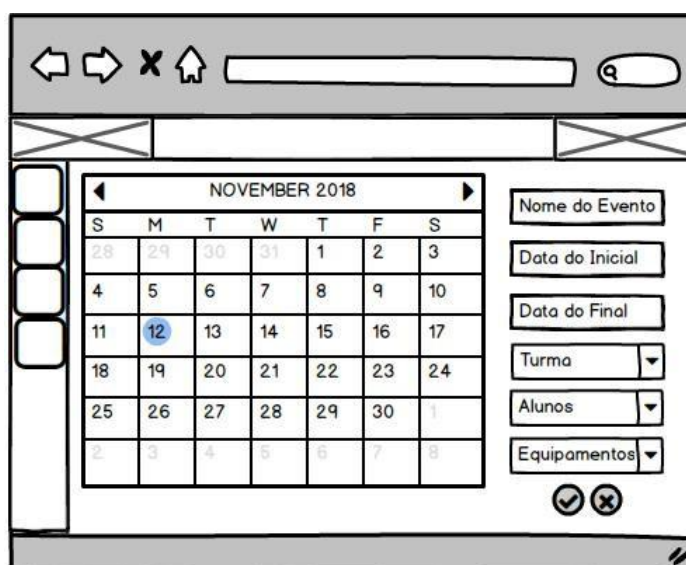


Figura 21 - Protótipo de tela de cadastro.
Fonte : Autoria Própria.

O cadastramento de reserva seguirá um padrão diferente de telas buscando tornar a interface e a usabilidade seu principal ponto. Nesta tela o usuário deverá ter a possibilidade de visualizar as reservas já existentes, permitir o cadastramento, assim como seu cancelamento. Nesta tela deverá seguir as regras de negócio especificadas para a geração de reservas como mostrado na Figura 22.



NOVEMBER 2018						
S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Nome do Evento

Data do Inicial

Data do Final

Turma

Alunos

Equipamentos

✓ ✕

Figura 22 - Protótipo da tela de cadastro de reservas.
Fonte : Autoria Própria.

Para os alunos terá telas especiais para visualizar suas reservas, juntamente com a possibilidade de utilizar o equipamento quando o mesmo estiver no tempo da reserva. Pode-se observar na Figura 23 o respectivo protótipo.

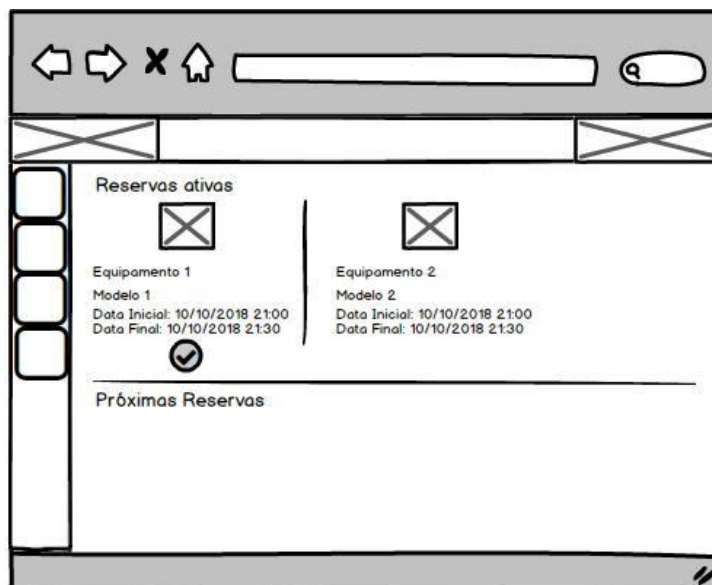


Figura 23 - Tela de visualização de reservas.
Fonte: Autoria Própria.

Finalmente o usuário aluno ou professor que tiver uma reserva, o sistema liberará uma opção para a realização dos comandos e treinamento com os equipamentos, o protótipo pode ser visualizado na Figura 24.

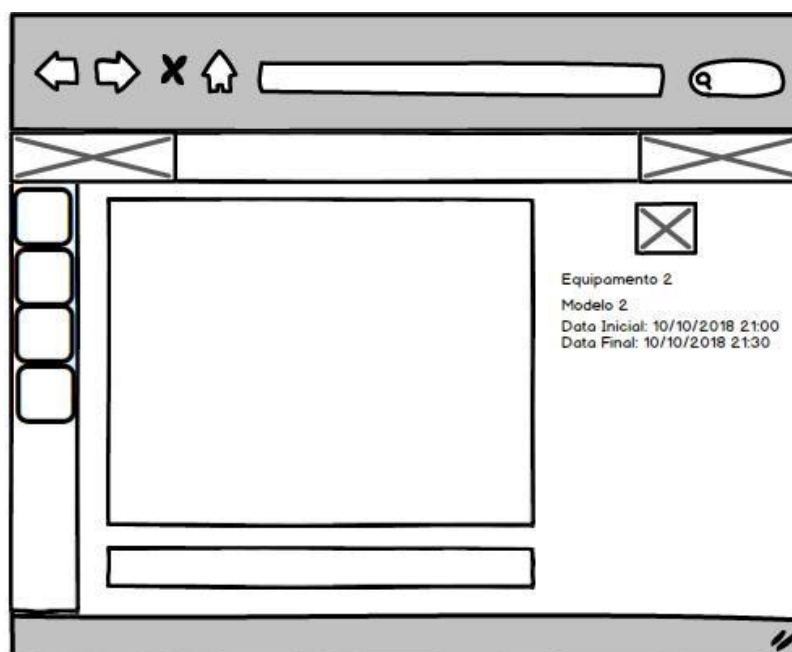


Figura 24 - Protótipo de tela de acesso via terminal.
Fonte: Autoria Própria.

5.2.6 Desenvolvimento do sistema.

Para o desenvolvimento do sistema foram utilizados os dados levantados por esse trabalho na fase de análise e levantamento de requisitos. Para o desenvolvimento foram utilizados: IDE de desenvolvimento VSCode; banco de dados Postgres; linguagem de programação python; linguagem de marcação de texto HTML5; framework Angular e Flask;

Para acessar o sistema será ser utilizado o endereço do site de visualização clicando em área restrita, será direcionado para tela de Login mostrado na Figura 25, que, só o usuário terá permissão para o acesso.

A imagem mostra a interface de login de um sistema web. No topo, há o logotipo da UTFPR (Universidade Tecnológica Federal do Paraná) em preto e amarelo, com o texto "UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ" abaixo dele. Abaixo do logotipo, o texto "Laboratório virtual de redes !" é exibido. O formulário de login contém dois campos de entrada: "Informe seu usuário" com o texto "cleiton@gmail" e "Informe seu senha" com pontos para ocultar a senha. Abaixo dos campos, há um botão preto com o texto "ACESSAR" e um ícone de lupa para pesquisa.

Figura 25 - Tela de login
Fonte: Autoria Própria.

O *login* é efetuado por meio do e-mail e senha gerada pelo professor ou administrador do sistema do projeto. Após digitado os campos necessários e clicado no botão para acesso, as informações serão verificadas, estando correto será direcionada para tela principal, conforme mostrado na Figura 26 do sistema de controle, caso esteja errado uma mensagem será exibida para o usuário.

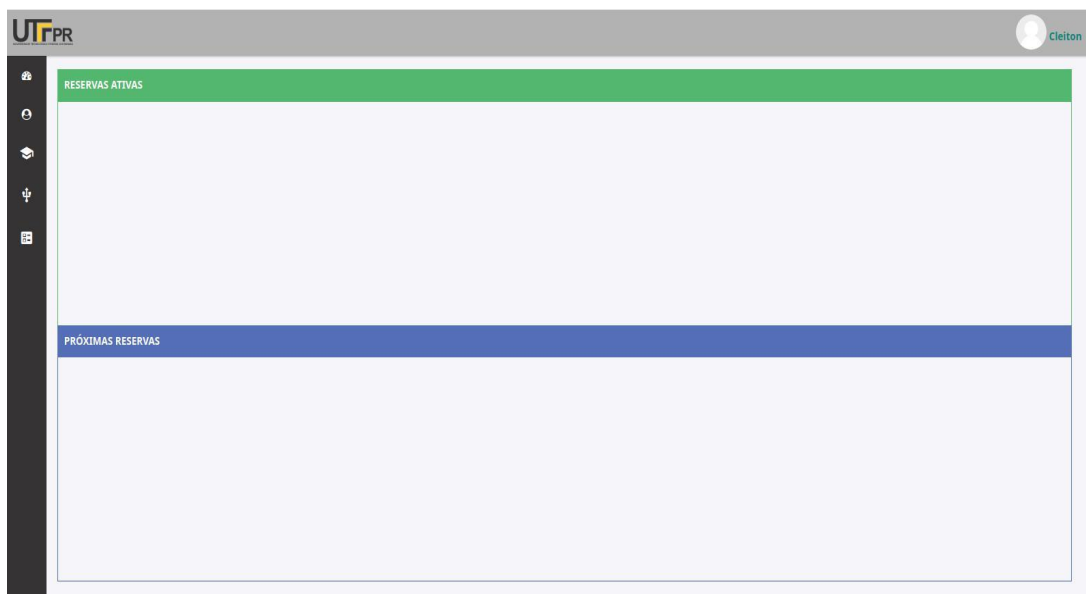


Figura 26 - Tela inicial do sistema.

Fonte: Autoria Própria.

A Tela inicial será personalizada pela permissão do usuário, levando em consideração as restrições de cada tipo, isso faz com que cada usuário tenha acesso a funcionalidades que são de suas responsabilidades.

A tela principal contará com um menu lateral, que terá com as opções de: usuários, turmas, equipamento e reservas

A navegação entre páginas é feita por meio de anotações e configurações utilizando as rotas do Angular conforme observadas na Listagem 1 .

```

const appRoutes: Routes = [
  { path : 'login', component : LoginComponent},
  { path : 'home', component: HomeComponent,
    children : [
      { path : 'consultaReservasUsuario', component :
ConsultaReservasAlunosComponent },
      { path : 'consultaUsuarios', component :
ConsultaUsuarioComponent },
      { path : 'consultaTurmas', component :
ConsultaTurmaComponent },
      { path : 'consultaEquipamentos', component :
ConsultaEquipamentoComponent },
      { path : 'consultaReservas', component :
ConsultaReservasComponent },
    ]},
  { path : '', pathMatch : 'full', redirectTo : 'login'}
];

```

Listagem 1 - Definição de rotas.

Fonte: Autoria Própria.

Essas rotas serão chamadas por meio do componente de meu lateral, construído com HTML5 e demarcado com as propriedades do Angular, mostradas na Listagem 2.

```

<a title ="Usuários" routerLink="consultaUsuarios">
  <i class="fa fa-user-circle"></i>
  <span class="">Usuários</span>
</a>

```

Listagem 2 - Chamada de rotas no HTML

Fonte: Autoria Própria.

Essas rotas serão adicionadas em um espaço definido pelo desenvolvedor, que irá ser carregado apenas o conteúdo requerido. O local deverá ser inserido e deve contar com a tag router-outlet.

5.2.6.1 Tela de cadastro de usuários.

As telas de cadastros, consultas e exclusões foram modeladas para melhor interação com o usuário, buscando sempre tornar o trabalho do usuário mais fácil. A Figura 27 mostra a tela de consulta, edição e

exclusão de usuários, lembrando que essas definições podem sofrer mudanças com as permissões do usuário.

ID	Nome	Email	Categoria	Situação	Turma	Ações
1	Cleiton	cleiton@gmail	Professor	Ativo	Turma 1	[Edit] [Deactivate] [Delete]
2	Joao Paulo	joao@gmail	Aluno	Ativo	Turma 1	[Edit] [Deactivate] [Delete]
3	Fabiano	fabiano@gmail.com	Aluno	Ativo	Turma 1	[Edit] [Deactivate] [Delete]
4	Roberto	roberto@gmail	Professor	Ativo	Turma 0	[Edit] [Deactivate] [Delete]
5	Carlos	carlos@carlos	Aluno	Ativo	Turma 1	[Edit] [Deactivate] [Delete]

Figura 27 - Tela de consulta de usuário.
Fonte: Autoria Própria.

Para buscar todos os usuários nesta tela o sistema utiliza uma comunicação via *web services* a uma API em python, essa comunicação utiliza do *Framework* Flask, na Listagem 3 é demonstrado como essa requisição é realizada pelo cliente.

```

buscarTodosUsuarios() {
    return this.http.doPost('buscarTodosUsuarios',
this.usuario).subscribe(retorno =>
    { if (retorno != null) {
        this.listaUsuario = retorno;
        return this.listaUsuario;
    }
    });
}

```

Listagem 3 - Requisição para API em python.
Fonte: Autoria Própria.

Já na listagem 4 mostra como essa requisição é recebida pelo lado do servidor.

```
app = Flask(_name_)
CORS(app)
@app.route("/buscarTodosUsuarios", methods=['POST'])
def buscarTodosUsuarios():
    return userWs.buscarTodosUsuarios();
```

Listagem 4 - API em python buscando relacionar a busca dos usuários.
Fonte: Autoria Própria.

Na Listagem 4 a marcação `@app.route` é a forma a qual o framework Flask demarca o método usado na requisição juntamente com qual será o nome e o formato que a API responderá para o cliente.

Nesta tela o usuário terá a possibilidade também de adicionar um novo usuário do sistema, e claro que a adição são restritas a usuários do tipo administrador e professor. Essa tela pode ser visualizada na Figura 28.

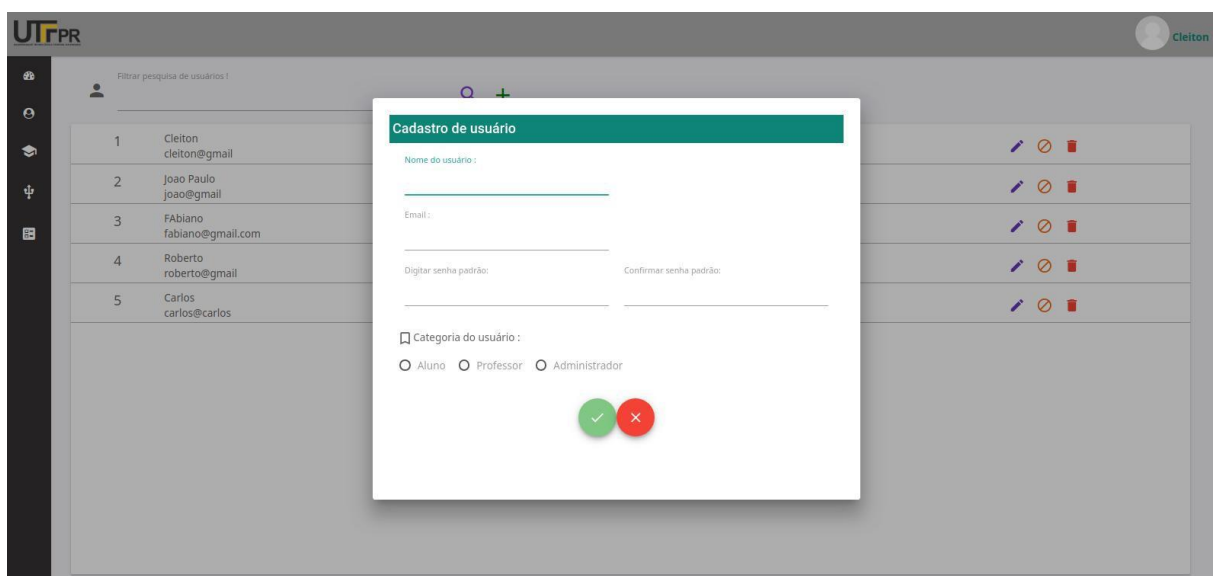
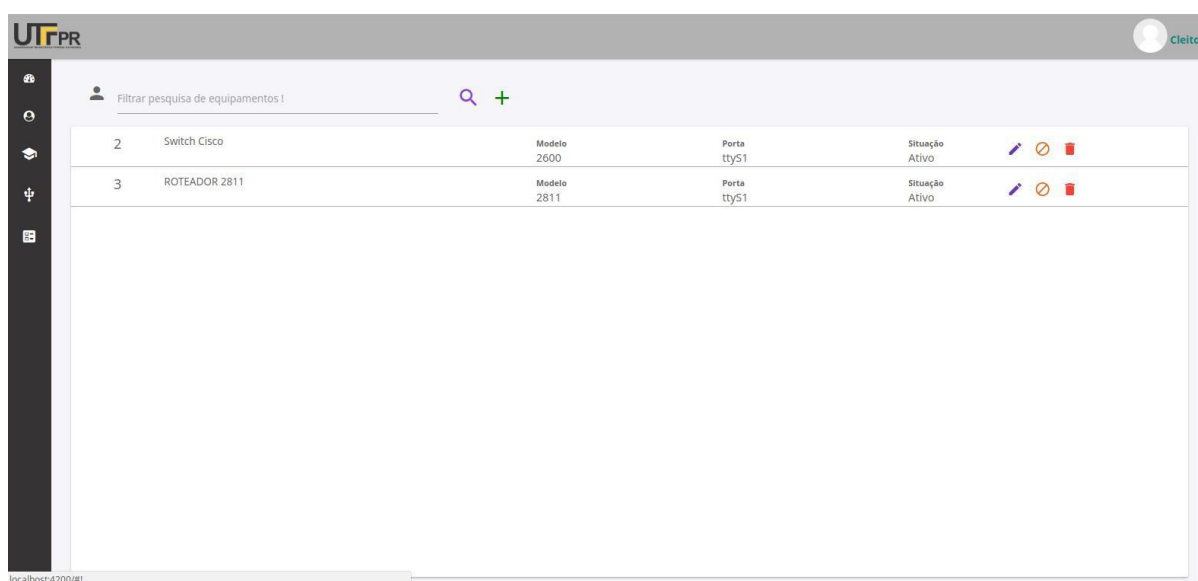


Figura 28 - Tela de cadastro de usuário.
Fonte: Autoria Própria.

5.2.6.2 Tela de cadastro de equipamentos

Nesta tela o usuário terá a possibilidade de adicionar quantos e qualquer equipamento que desejar, criando assim, um cenário de rede o qual pode ser modularizado conforme a necessidade dos envolvidos. Para adicionar um equipamento novo o usuário precisa saber informar em qual porta do equipamento está ligado com o servidor e claro que a adição são restritas a usuários do tipo administrador. Essa tela pode ser visualizada na Figura 29.



ID	Nome	Modelo	Porta	Situação	Ações
2	Switch Cisco	2600	tty51	Ativo	[Editar] [Excluir] [Atualizar]
3	ROTEADOR 2811	2811	tty51	Ativo	[Editar] [Excluir] [Atualizar]

Figura 29 - Tela de consulta de equipamento.
Fonte: Autoria Própria.

Na Figura 30 pode-se visualizar o cadastro dos equipamentos e os campos que são necessários para o cadastramento de um equipamento.

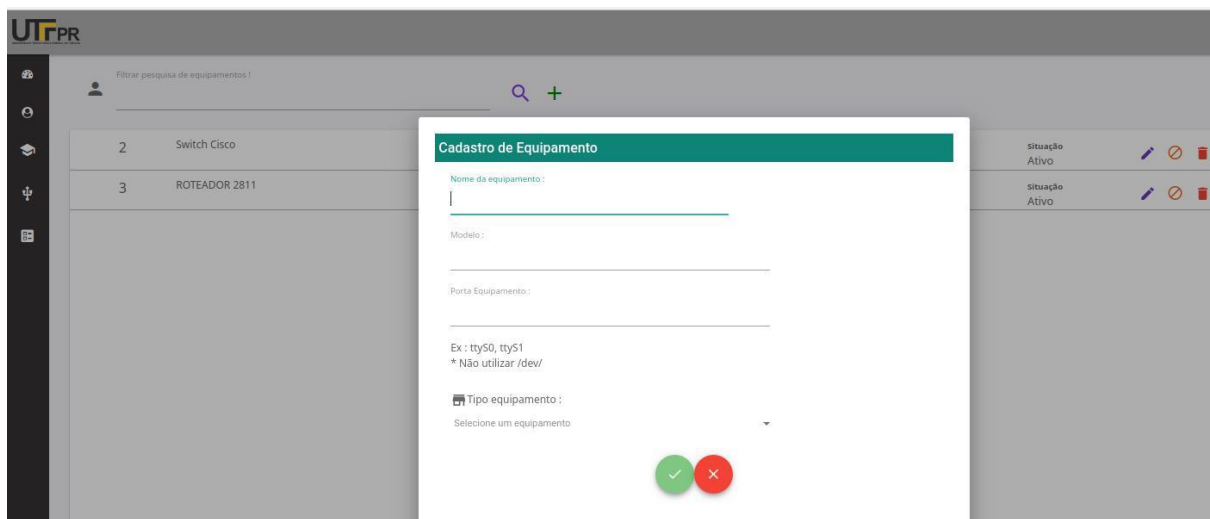


Figura 30 - Tela de cadastro de equipamento.
Fonte: Autoria Própria.

5.2.6.3 Tela de cadastro de reservas.

A principal funcionalidade que será gerenciada pelo sistema será o controle, cadastramento e o cancelamento da reserva, tela mostrada na Figura 31. O sistema em questão trata a possibilidade da geração de reserva individual o para uma turma inteira, desde que, no período requisitado não tenha reserva para outros usuários.

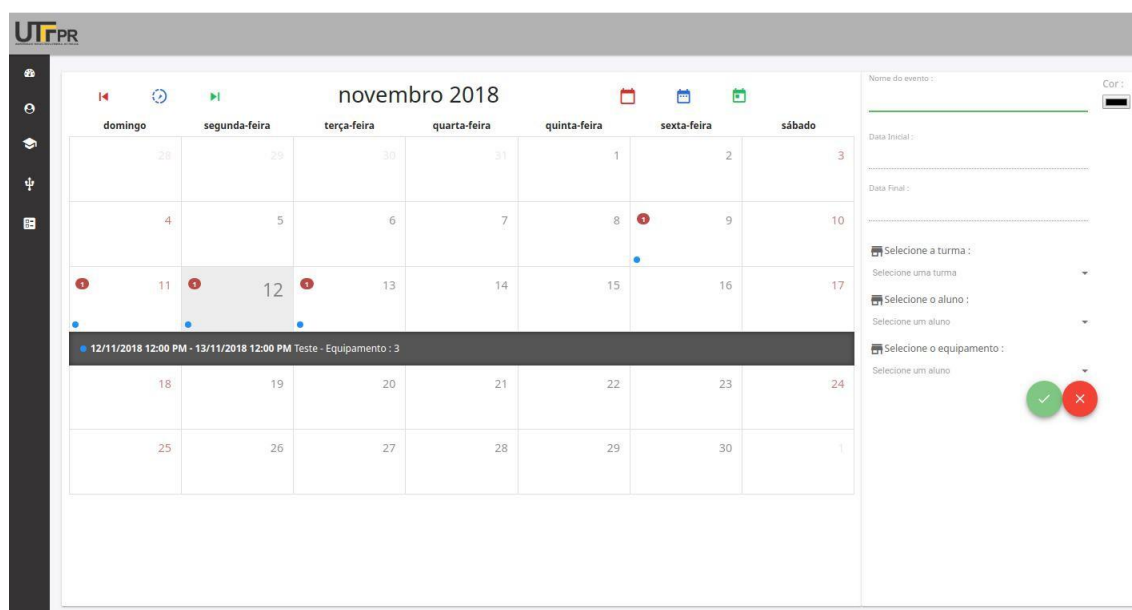


Figura 31 - Tela de reserva.
Fonte: Autoria Própria.

5.2.6.4 Consulta de reservas e utilização dos equipamentos.

Mediante a uma reserva o aluno poderá realizar o acesso a equipamentos que estão previamente reservados para o mesmo, o mesmo ao realizar o login no sistema já terá apresentado na tela inicial quais são suas reservas ativas e suas próximas reservas, como mostrado na Figura 32.

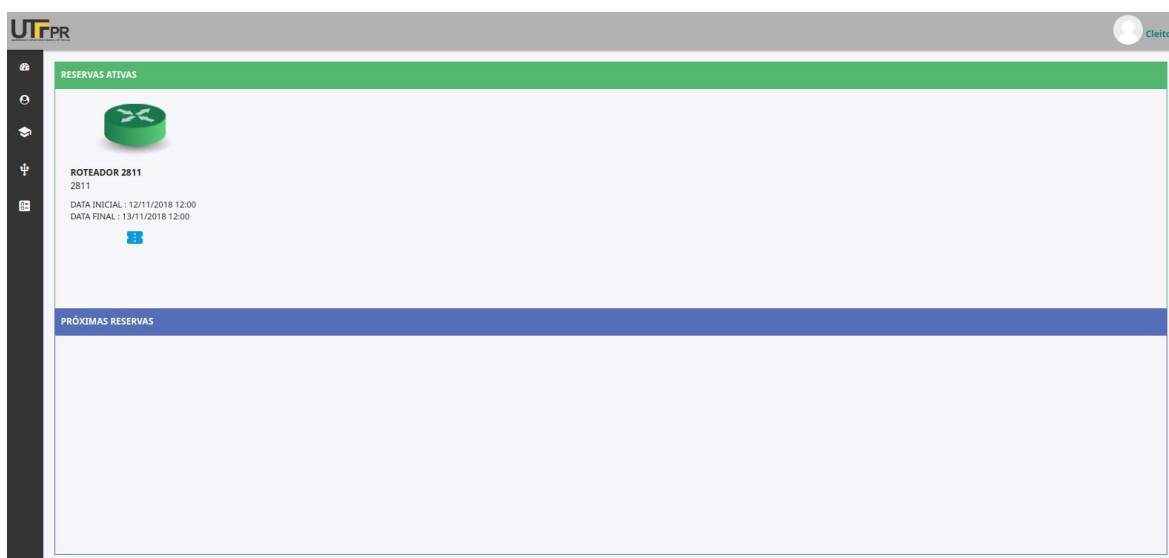


Figura 32 - Tela de consulta de reserva por parte do aluno.
Fonte: Autoria Própria.

Ao selecionar uma reserva que está ativa o acadêmico será direcionado para a tela mostrada na Figura 33, que o possibilitará manipular o equipamento pela plataforma.

6 CONCLUSÕES

Para todas as matérias de cursos de tecnologias que envolvem redes de computadores, sempre existe a necessidade de realizar as atividades e aplicar os conceitos vistos em sala em um ambiente prático voltado a fixação. Esses pontos fazem com que o uso de equipamentos se tornem indispensáveis no desenvolvimento profissional de cada acadêmico.

Com a evolução do conhecimento dos acadêmicos sendo ligada totalmente com a experiência vivida no ambiente acadêmico, existe a necessidade de aproximar os acadêmicos com um ambiente que seja o mais próximo que o mesmo vai encontrar em um mundo real, isso faz com que o uso, a manipulação e a interação do aluno com equipamentos forneça uma eficiência maior em aulas e no desenvolvimento acadêmico.

O projeto desenvolvido busca auxiliar na aprendizagem das disciplinas que estão relacionadas a redes de computadores e telecomunicações, para isso, foi desenvolvido um sistema web que proporciona o acesso remoto a equipamentos. Este projeto busca auxiliar, acadêmicos e professores, na solução de resolver problemas que são enfrentados em um laboratório de redes.

A utilização do software proposto busca dividir as finalidades de cada usuário, deixando seu uso personalizado e adequado para cada usuário, criando as atividades de modo que cada usuário tenha acesso restrito às suas funções. Também ajudará no quesito uso e reservas de equipamento, pois conta com um serviço onde os professores podem realizar a reserva de equipamentos para alunos ou reservas para turmas inteiras de modo intuitivo e fácil.

Ao fim do processo de desenvolvimento, conseguiu-se concluir os objetivos propostos por este trabalho na forma de um sistema WEB, que auxiliará na gerência e na utilização de laboratório de redes de forma remota e virtual.

Visando que todo software pode ser melhorado, será possível a implementação de melhorias no sistema, visando sempre o auxílio no

aprendizado dos acadêmicos que buscam aperfeiçoar seu conhecimento nas disciplinas de redes de computadores.

REFERÊNCIAS

ALENCAR, M. A. DOS SANTOS, **FUNDAMENTOS DE REDES DE COMPUTADORES**. Manaus-AM, 2010.

ANGULAR DOCS. Disponível em <https://angular.io/docs>. Acesso em 14 Setembro. 2018.

BARBOSA, Simone D. Junqueira; SILVA, Bruno Santana. **Interação Humano-Computador**. Rio de Janeiro: Elsevier, 2010.

BENYON, D. **Designing Interactive Systems: A Comprehensive Guide to HCI, UX & Interaction Design**, 3rd ED, 2011.

CANTELON, M. et al. **Node.js in Action**. 1 ed. Manning Publications, 2014.

CANTU, E. **Redes de computadores e internet**. São José-SC: CEFET/SC, 2003.

Diagramas Comportamentais. Disponível em goo.gl/3wdH17. Acesso em 07 Setembro de 2018.

DUBOIS. P.F. **Python: Batteries Included, volume 9 of Computing in Science & Engineering**. IEEE/AIP. 2007.

FLATO, P. A. U; GUIMARÃES; P. H. **Educação baseada em simulação em medicina de urgência e emergência: a arte imita a vida***, 2011.

FLASK. Disponível em <https://flask.pocoo.org>. Acesso em 20 Setembro. 2018.

FEREIRA, K. H. A.; DE LIMA, R. W.; CHAVES, J. O.; LIMA, A. **Inserindo um Laboratório Virtual para o Ensino de Redes de Computadores** Disponível <<http://www.icbl-conference.org/proceedings/2013/papers>>. Acesso em 7 Setembro de 2018.

FILIPPETTI, Marcos A. **CCNA 4.1 Guia Completo de Estudo.**, 2011.

FOROUZAN, A. Behrouz. **Comunicação de Dados e Redes de Computadores**. Editora AMGH, 2010.

FRANCISCATTO, R; CRISTO, F; PERLIN, T. **Redes de Computadores**. UFSM, 2014.

GNS3. **The software that empowers network professionals**. Disponível em < <https://www.gns3.com/>>. Acesso em: 15 de outubro de 2018.

GOMES, L.; BOGOSYAN, S. Current Trends in Remote Laboratories. **Ieee Transactions On Industrial Electronics**, [s.l.], v. 56, n. 12, p.4744- 4756, dez. 2009. Institute of Electrical and Electronics Engineers (IEEE). Disponível em <<http://dx.doi.org/10.1109/tie.2009.2033293>>. Acesso em 7 Setembro de 2018.

GUERBER, C. **PROTOCOLO IP**. Universidade do Contestado, Mafra-SC, 2007.

HADDAD, R. **Web Services**. Disponível em <https://msdn.microsoft.com/pt-br/library/cc564893.aspx>. Acesso em 09 Setembro. 2018.

KNEBEL, E; BARRISTI, G. **Simulando redes complexas com o GNS3**. Departamento de Ciências Exatas e Engenharias, 2012.

KRUCHTEN, P. **The rational unified process: An introduction**. Boston: Pearson Education, Inc, 2004.

KURUSE, J. F; ROSS, K. W. **Redes de computadores e a Internet: Uma nova abordagem**, Addison Wesley, São Paulo, 2003.

LAPERUTA, D.; SILVA, L.; MIKAMI, R. **Implementação de um laboratório de acesso remoto para apoio às aulas práticas das disciplinas de Redes de Computadores**. Monografia (Curso de Tecnologia em sistema para a internet) - Universidade tecnológica federal do Paraná departamento Acadêmico de Informática, Curitiba, 2011.

MARTINS, C. V. **Sistema para almoxarifado**, FUNDAÇÃO EDUCACIONAL DO MUNICÍPIO DE ASSIS, 2014.

NEDIC, Z; MACHOTKA, J; NAFALSKI, A. **Remote laboratory netlab for effective interaction with real equipment over the internet**. 2008 Conference on Human System Interactions, Krakow, 2008, pp. 846-851.

NETLAB+. Disponível em <https://www.netdevgroup.com/content/cnap/lab/s/>. Acesso em 20 Setembro. 2018.

OLIVEIRA, D. T. de; CORTIMIGLIA, M. N.; LONGHI, M. T. **Ambientes virtuais de aprendizagem no ensino superior presencial: o processo de adoção da tecnologia na perspectiva do docente**. RBAAD, v. 14, p. 37-54, 2015. Disponível em: . Acesso em: 7 agosto. 2018.

PACKETTRACER, **Cisco Packet Tracer**. Disponível em <https://www.netacad.com/pt-br/courses/packet-tracer>. Acesso em 05 Outubro 2018.

PIRES, J. **UML, Diagramas estruturais.** Disponível em <http://www.slideshare.net/thaisedd/uml-diagramas-estruturais-parte-escrita>. Acesso em 07 Setembro. 2018.

PRESSMAN, Roger. **Engenharia de software.** São Paulo: Makron Books, 2008.

Prototipação de interfaces: o PROFACE. Disponível em <http://zeluisbraga.wordpress.com/2007/04/11/prototipacao-de-interfaces-o-profaced/>. Acesso em 07 Setembro. 2018.

SEABRA, M. RODOLFO. **Análise e projeto orientado a objeto usando UML e o processo unificado.** Disponível em <http://www.ufpa.br/cdesouza/teaching/cedai/APOOUMLP.pdf> Acesso em 07 Setembro de 2018.

SOUZA, M; SILVA, J; et al. **Simuladores e Emuladores de Rede para o Projeto e Solução de Problemas em Ambientes de Produção.** Revista de tecnologia da informação e comunicação, VOL. 6, 2016

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de computadores.** 5ª edição. São Paulo, 2011.

TANENBAUM, A. S. **Redes de Computadores.** Amsterdam, Holanda. Tradutor: VANDENBERG D. de Souza, Campus 2003.

UML. Disponível em <http://www.cassao.eti.br/portal/unified-modeling-language> . Acesso em 07 Setembro. 2018.

VISÃO GERAL DO HTML5. Disponível em <http://tableless.com.br/html5/?chapter=1>. Acesso em 07 Novembro. 2013.
Web Services. Disponível em <http://msdn.microsoft.com/pt-br/library/cc564893.aspx>. Acesso em 08 Setembro. 2018.