

**UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**SISTEMA DE GERENCIAMENTO E CONTROLE DE SERVIDORES DA
MIKROTIK**

LEANDRO GONÇALVES DE ARAÚJO

**PICOS – PIAUÍ
2016**

LEANDRO GONÇALVES DE ARAÚJO

**SISTEMA DE GERENCIAMENTO E CONTROLE DE
SERVIDORES DA MIKROTIK**

Monografia submetida ao Curso de Bacharelado de Sistemas de Informação como requisito parcial para obtenção de grau de Bacharel em Sistemas de Informação.

Orientador (a): Prof.^a Patricia Medyna Lauritzen de Lucena Drumond.

**PICOS – PIAUÍ
2016**

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

A659s Araújo, Leandro Gonçalves de.

Sistema de gerenciamento e controle de servidores da
Mikrotik / Leandro Gonçalves de Araújo.– 2016.

CD-ROM : il.; 4 ¾ pol. (42 f.)

Monografia (Curso Bacharelado em Sistemas de Informação) –
Universidade Federal do Piauí, Picos, 2015.

Orientador(A): Profª. Ma. Patrícia Medyna Lauritzen de Lucena
Drumond

1. Redes de Computadores. 2. Sistema *Web*. 3. *Mikrotik RouterOS*. I. Título.

CDD 004.69

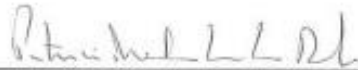
SISTEMA DE GERENCIAMENTO E CONTROLE DE SERVIDORES DA MIKROTIK

LEANDRO GONÇALVES DE ARAÚJO

Monografia aprovada como exigência parcial para obtenção do grau de
Bacharel em Sistemas de Informação.

Data de Aprovação

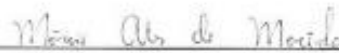
Picos - PI, 22 de fevereiro de 2016



Prof. Ma. Patricia Medyna Lauritzen de Lucena Drumond
Orientadora



Prof. Esp. Pablo de Abreu Vieira
Membro



Prof. Me. Márcio Alves de Macêdo
Membro

AGRADECIMENTOS

Agradeço a minha Mãe, pois foi seu cuidado e dedicação que me fez seguir em frente nesta batalha para que um dia possa retribuir todos os esforços e abdições que a senhora se prestou para ver seus filhos felizes. A meu Pai pessoa que admiro e tenho como exemplo de vida, foi ele que me ensinou o valor do trabalho e seguir sempre em busca de uma vida melhor.

Também não posso me esquecer dos meus irmãos Francisca e Leonardo, presentes em cada etapa da minha vida, participando dos momentos bons e ruins.

Aos professores que me acompanharam durante a graduação, vocês são os responsáveis por grande parte do conhecimento que tenho hoje, em especial a Professora, Patrícia Medyna, que me orientou na elaboração desta monografia.

Agradeço imensamente a você minha namorada, Patrícia, que participou desse momento comigo, compreendendo minhas ausências. Obrigado pelo carinho e por sua capacidade de me trazer paz na correria de cada semestre.

E o que dizer dos meus amigos e colegas que convivi em sala de aula, os funcionários do CTI, foram eles que por diversas vezes dedicaram um minuto de seu tempo para tirar dúvidas e compartilhar um pouco de seu conhecimento, a vocês só tenho a agradecer.

Enfim obrigada meu Deus por estar sempre guiando meus passos e me protegendo de todo mal, se não fosse pela tua vontade não mais estaria aqui.

"Determinação coragem e autoconfiança são fatores decisivos para o sucesso. Se estamos possuídos por uma inabalável determinação conseguiremos superá-los.

Independentemente das circunstâncias, devemos ser sempre humildes, recatados e despidos de orgulho."

(Dalai Lama)

RESUMO

A cada dia que passa o número de pessoas com o acesso à *internet* cresce de maneira exponencial, conforme pesquisas realizadas pelo IBGE - Instituto Brasileiro de Geografia Estatística - em maio de 2013 esse número aumentou cerca de 143,8% de 2005 a 2011. Entretanto, a medida em que ocorre o crescimento das redes de computadores, surge a necessidade de se desenvolverem novas tecnologias, aumentando assim a viabilidade de interconexão entre os usuários. Segundo a ABRINT (2014) (Associação Brasileira de Provedores de Internet e Telecomunicação), cerca de 22 % do acesso à *internet* hoje no Brasil é feita através de pequenos provedores regionais, de modo que na maioria dos pequenos e médios provedores de serviço de *internet* usam o sistema operacional de rede da *Mikrotik* o *RouterOS*. Diante do exposto o presente trabalho científico tem como objetivo desenvolver uma aplicação capaz de gerenciar servidores que trabalham com o sistema *RouterOS*, para facilitar a configuração e manutenção dos equipamentos de rede, obtendo assim maior facilidade no gerenciamento dos mesmos. Para que os resultados sejam obtidos, será necessário o uso de tecnologias como: a linguagem de programação *Python*, o *framework Django*, assim como um conjunto de outras tecnologias de desenvolvimento *web*, também torna-se necessário o conhecimento do sistema operacional *RouterOS*. As metodologias e ferramentas utilizadas, e as conclusões obtidas serão explicitadas durante o desenvolvimento do trabalho.

Palavras-chave: Redes, sistema *web*, *mikrotik*, *routers*, *python*, *django*.

ABSTRACT

With each passing day the number of people with access to the Internet is growing exponentially, according to surveys conducted by IBGE – Instituto Brasileiro de Geografia e Estatística - May 2013 this number increased by approximately 143.8% from 2005 to 2011. However the extent that occurs the growth of computer networks, there is a need to develop new technologies, thereby increasing the viability of interconnection between users. According to ABRINT (2014) (Associação Brasileira de Provedores de Internet e Telecomunicação), about 22% of Internet access today in Brazil is done through small regional providers, so that in most small and medium-sized Internet service providers. They use the network operating system of the Mikrotik RouterOS. Given the above the present scientific study aims to develop an application that can manage servers that work with RouterOS system, to facilitate the configuration and maintenance of network equipment, thus obtaining greater ease in managing them. For the results to be obtained, the use of technology will be required as the programming language Python, Django framework, as well as a number of other web development technologies also becomes necessary to know the RouterOS operating system. The methodologies and tools used, and findings will be explained during the development of this work.

Keywords: networks, web system, mikrotik, RouterOS. Python, Django.

LISTA DE ILUSTRAÇÕES

Figura 1 - Ciclo de vida da aplicação	25
Figura 2 - Terminal do RouterOs.....	30
Figura 3 - Ferramenta de Winbox.....	30
Figura 4 - Configuração de redirecionamento pelo WinBox	31
Figura 5 - Redirecionamento via Terminal.....	31
Figura 6 - Diagrama de Caso de Uso	32
Figura 7 - Diagrama de Classe Autenticação	33
Figura 8 - Diagrama de Classe.....	33
Figura 9 - Tela de Autenticação	34
Figura 10 – Tela Inicial	34
Figura 11 - Cadastro de Servidores	35
Figura 12 - Lista de Servidores	35
Figura 13 - Cadastro de Planos.....	36
Figura 14 - Cadastro de Clientes.....	36
Figura 15 - Tela de Pagamento.....	37

LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais	27
Quadro 2 - Requisitos não Funcionais.....	28

LISTA DE ABREVIATURAS E SIGLAS

ABRINT	Associação Brasileira de Provedores de <i>Internet</i> e Telecomunicação
AP	<i>Access Point</i>
BGP	<i>Border Gateway Protocol</i>
CWI	<i>Centrum Wiskunde & Informatica</i>
DNS	<i>Domain Name System</i>
GPL	<i>General Public License</i>
IBGE	Instituto Brasileiro de Geografia Estatística
ISP	<i>Internet Service Provider</i>
NAT	Network Address Translation
OMT	<i>Object-Modeling Technique</i>
OOSE	<i>Object-Oriented Software Engineering</i>
ORM	<i>Mapping Object-Relationa</i>
OSPF	<i>Open Shortest Path First</i>
PBM	Pesquisa Brasileira de Mídia
PPP	<i>Point to Point Protocol</i>
PPPoE	<i>Point-to-Point Protocol over Ethernet</i>
RTCP	<i>Real-Time Control Protocol</i>
RIP	<i>Raster Imagem Processor</i>
RTP	<i>Real-time Transport protocol</i>
SOL	Sistemas operacionais locais
SOR	Sistema operacional de rede
SSH	<i>Secure Shell</i>
SSHv2	<i>Secure Shell version 2</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
TI	Tecnologia da informação
VOIP	<i>Voice over Internet Protocol</i>
VRRP	<i>Virtual Router Redundancy Protocol</i>
WDS	Wireless Distribution System
WWW	<i>Word Wide Web</i>

SUMÁRIO

1. INTRODUÇÃO	10
1.1. OBJETIVOS	11
1.2. ORGANIZAÇÃO DO TRABALHO	11
2. REFERENCIAL TEÓRICO	11
2.1. SISTEMAS OPERACIONAIS DE REDE	12
2.2. SERVIDORES <i>LINUX</i>	13
2.2.1.RouterOs	13
2.3. SISTEMA DE INFORMAÇÃO	16
2.4. SISTEMAS WEB	18
2.5. TECNOLOGIA DE INFORMAÇÃO.....	19
2.6. LINGUAGUENS DE PROGRAMAÇÃO.....	20
2.7 <i>PYTHON</i>	20
2.8 <i>DJANGO</i>	23
2.9 ENGENHARIA DE <i>SOFTWARE</i>	23
3. ETAPAS E FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO DO SCG-MK- SISTEMA DE GERENCIAMENTO E CONTROLE DE SERVIDORES DA MIKROTIK.	26
3.2. ENTENDENDO O SISTEMA.....	32
3.3. TELA DE LOGIN	33
3.4. TELA INICIAL.....	34
3.5. CADASTRO DE SERVIDORES	34
3.6. LISTA DE SERVIDORES	35
3.7. CADASTRO DE PLANOS	35
3.8. CADASTRO DE CLIENTES	36
3.9. TELA DE PAGAMENTO.....	36
4. CONSIDERAÇÕES FINAIS	37
5. REFERÊNCIAS BIBLIOGRÁFICAS	39

1. INTRODUÇÃO

Nos dias de hoje, as Tecnologias de Informação (TI) estão cada vez mais presentes na vida das pessoas, as TIs são responsáveis por tornar eficiente a maneira de trabalhar e desenvolver as atividades, sejam estas no ambiente de trabalho ou na vida privada.

A *internet* é um vasto conjunto de redes diferentes que utilizam certos protocolos comuns e fornecem determinados serviços comuns. É um sistema pouco usual no sentido de não ter sido planejado, nem ser controlado por ninguém e os elementos que formam sua base são o modelo de referência TCP/IP (*Transmission Control Protocol / Internet Protocol*) e a pilha de protocolos TCP/IP, na qual possibilita a criação de um serviço universal (TANENBAUM, 2003).

De acordo com portal Brasil (2014), cerca de 48% da população brasileira usa a *internet* regularmente, é o que mostra a Pesquisa Brasileira de Mídia (PBM). Além disso o percentual de pessoas que fazem uso diariamente cresceu de 26% na PBM 2014 para 37% na PBM 2015. Com este crescente número de usuários houve a necessidade de surgirem novas tecnologias e aplicações capazes de gerenciar e controlar o ambiente de redes, além de garantir a qualidade de serviço QoS (*Quality of Service*).

Ao longo dos últimos anos os pequenos provedores ganharam espaço no mercado de transmissão de dados levando acesso aos lugares mais longínquos, não atingidos por grandes empresas. Em muitos casos esses pequenos provedores de serviços não se deparam com uma grande estrutura e equipamentos de grande porte, porém são capazes de levar *internet* aquelas localidades menos favorecidas.

Atualmente no Brasil grande parte dos pequenos provedores de serviço de *internet* usam *RouterOs* como servidor. Este sistema operacional foi desenvolvido para criar solução em conectividade como ISP (*Internet Service Provider*), administração de rede e serviço de conexão sem fio, além de operar similarmente em computadores baseados em plataforma x86.

Diversas soluções de conectividade podem ser implementadas através do *RouterOs*, possuindo este, ferramenta própria de configuração chamada *Winbox*, fornecendo aos administradores de rede total acesso ao sistema. As configurações

podem ser feita via SSH (*Secure Shell*) ou TELNET, sendo os comandos enviados via terminal.

1.1.OBJETIVOS

O objetivo geral deste projeto foi a criação de uma ferramenta computacional com propósito de abstrair grande parte da complexidade de configuração do *RouterOs*, tornando-o um ambiente amigável, para que o usuário possa melhor utilizar dos recursos do sistema operacional, de modo que possa fornecer melhorias na realização das configurações de redes e *firewall*, agilidade nos cadastros de clientes, facilitar os modos de conexão, proporcionando assim uma excelente administração dos gerentes de rede.

1.2. ORGANIZAÇÃO DO TRABALHO

O trabalho desenvolvido será apresentado nos próximos capítulos organizado da seguinte maneira:

- Capítulo 2 – Referencial Teórico: Neste capítulo é abordado todo o embasamento teórico obtido durante o desenvolvimento do trabalho
- Capítulo 3 - Etapas E Ferramentas Utilizadas Para O Desenvolvimento Do Scg-Mk- Sistema De Gerenciamento E Controle De Servidores Da *Mikrotik*: Neste capítulo são descritas as etapas e as ferramentas para o desenvolvimento do sistema.
- Capítulo 4 – Considerações Finais: Esse capítulo apresenta os conhecimentos adquiridos e os problemas enfrentados no decorrer do desenvolvimento. Também são explanadas as ideias e dicas para trabalhos futuros, afim de melhorar e dar sequência ao que foi desenvolvido nesse projeto.

2. REFERENCIAL TEÓRICO

A *internet* é uma rede que conecta computadores do mundo inteiro, criada em 1969 como projeto militar, em plena Guerra Fria. A princípio o propósito do governo americano era a comunicação segura, de forma a coibir possíveis ataques que viessem destruir os meios convencionais de telecomunicações. Além disso a rede mundial de computadores inicial foi criada também para disponibilizar aos centros de pesquisa das universidades, troca de informações de maneira rápida e segura (MENDES, 2007).

Atualmente este conceito não mais se adequa a nova realidade, sendo a mesma utilizada tanto para troca de informações seguras como meio de pesquisa, interação, negócios, enfim possui uma infinidade de utilidades essenciais para o dia-a-dia.

A *internet* é um conjunto de redes diferentes, que utilizam diversos meios de tecnologias, conectadas entre si, através de protocolos em comum, sua comunicação é definida principalmente pelos protocolos TCP e pelo IP, que formam o modelo de referência TCP/IP (TANENBAUM, 2003).

A proliferação da rede é uma consequência dos avanços da banda larga e os diversos dispositivos capazes de navegar pela *internet* como celulares, *tablets* televisores, videogames entre diversos outros dispositivos. Deste modo é preciso que se evolua, não apenas criando novas tecnologias, como também aperfeiçoando as já existentes, Mazza (2012) destaca que:

Os profissionais em grandes empresas como *Google, Microsoft, Apple e Facebook*, trabalham exclusivamente em melhorias para os navegadores mais utilizados, além de disseminar conhecimentos em grande comunidade de desenvolvedores.

2.1. SISTEMAS OPERACIONAIS DE REDE

Um Sistema Operacional de Rede (SOR) é um conjunto de módulos que se alarga dos sistemas operacionais, levando assim transparência aos recursos compartilhados da rede (BRANDT,2006).

Os SORs é uma espécie de aperfeiçoamento dos Sistemas Operacionais Locais (SOL), para que se possa torna perceptível a utilização dos recursos compartilhados.

De acordo com Pinheiro (2008), as modificações no hardware a favor da rede provocaram mudanças nos sistemas operacionais, de modo que precisaram se adaptar ao novo ambiente de processamento. Os computadores pessoais já funcionavam de forma insólita, onde os mesmos já estavam equipados com seus respectivos SOL.

Com o passar do tempo, surgiram os sistemas operacionais de rede com extensões dos sistemas locais, completando o conjunto de funções básicas e de uso geral tornando transparente o uso dos recursos compartilhados no sistema operacional. Uns dos requisitos principais dos SOR's é a transparência, onde esses atuam de forma que os usuários utilizem os recursos da rede como se estivesse operando localmente.

2.2. SERVIDORES *LINUX*

O *Linux* é *software* livre¹ construído a partir de *kernel*² do *UNIX*, projetado inicialmente para operar como sistema operacional de redes, porém é utilizado atualmente em diversos dispositivos tais como computadores, celulares e outros sem deixar de lado sua característica principal. O *Linux* é caracterizado por Miranda (2008) como:

Um sistema operacional de rede, onde apresenta um conjunto de módulos que ampliam as tarefas dos sistemas operacionais locais, complementando-os com um conjunto de funções básicas, e de uso geral, que tornam transparente o uso de recursos compartilhados da rede.

Como se trata de um *software* livre as mudanças do sistema operacional podem ser feitas de forma que resultem na criação de sistemas personalizados com finalidade específica a exemplo do *Ubuntu Server*, *Debian Server*, esses sistemas são os mais utilizados como servidores de aplicação web, podemos aqui destacar também como exemplo o *RouterOs*, este, porém foi totalmente modificado com finalidade de realizar a tarefa de um roteador.

2.2.1. *RouterOs*

¹ Permite-se adaptações ou modificações em seu código de forma espontânea

² Núcleo que gerencia os recursos do sistema

De acordo com sua documentação oficial o *RouterOs* é um sistema operacional de rede desenvolvido pela empresa *Mikrotik*, com a sua primeira versão no ano de 1995 (*Mikrotik*, 2015). Atualmente o *RouterOs* tem grande aceitação no mercado de Tecnologia de Informação por suas inúmeras funcionalidades. Além disso, trata-se de um sistema bem robusto e de grande estabilidade. Baseado em *Linux*, o sistema pode ser utilizado tanto em plataforma x86, como pode vir embarcado em *Hardware* próprio da empresa *Mikrotik* conhecido como *Routerboard*.

Algumas das funcionalidades que o *RouterOs* se destaca são:

- Performance otimizada com o protocolo proprietário *NSTREME*.
- Alta disponibilidade com o protocolo VRRP (*Virtual Router Redundancy Protocol*)
- Possibilidade de agregar *interfaces* físicas e virtuais
- Poucas exigências de recursos de *hardware*
- Qualidade de serviço avançado
- *Firewall "stateful"*³
- Protocolo *Spanning Tree*⁴ em *bridge* com filtros.
- Alta velocidade com 802.11a/b/g com criptografia WEP/WPA
- WDS e AP's Virtuais
- Portal Captativo (*Hotspot*) com acesso *Plug & Play*
- Roteamento com os protocolos RIP, OSPF e BGP
- Acesso remoto com amigável aplicativo *Windows* – *Winbox* e também administração WEB
- Administração por *telnet*, *mac-telnet*, *ssh* e *console*
- Configuração e monitoramento em tempo real

O *RouterOs* é bastante aceito tanto por aqueles que estejam iniciando seu aprendizado na área de redes como profissionais que já possuem sólidos conceitos na área, encontrando no *RouterOs* recursos para aumentar significativamente a produtividade e a qualidade de serviço, utilizando assim o máximo que o sistema oferece.

³ Firewall de estado capaz de identificar qual o protocolo os dados transitam.

⁴ Protocolo que trabalha com a topologia de anel, auxiliando uma melhor performance capaz de resolver problemas de loop na rede.

O *Domain Name System* (DNS) é um serviço de resolução de nomes que trabalha de forma hierárquica, que permite que ligue nomes de domínio particular com os respectivos endereços IP. A documentação oficial do *Mikrotik* (2015) fala que:

O *RouterOs*, tem recursos de servidor de DNS incorporado em cache DNS, assim, este recurso também pode ser usado para fornecer informações de DNS falso para seus clientes de rede. Por exemplo, resolver qualquer pedido DNS para um determinado conjunto de domínios (ou para toda a Internet) para sua própria página.

O servidor é capaz de resolver as solicitações de DNS com base em expressões regulares básicos, de modo que várias solicitações podem ser combinadas com a mesma entrada. No caso de uma entrada não está em conformidade com padrões de nomeação do DNS.

O protocolo *Point-to-Point Protocol over Ethernet* (PPPoE) fornece amplo gerenciamento de usuários, rede gestão e compatibilidade benefícios para ISPs e administradores de rede, o PPPoE é usado principalmente por ISPs para controlar as conexões de clientes além de ser uma extensão do padrão PPP (*Point to Point Protocol*), protocolo esse desenvolvido para permitir acesso autenticado, além de permitir pacotes de diversos protocolos. Em geral, PPPoE é utilizado para distribuir endereços IP aos clientes com base em autenticação por nome de usuário.

O *hotspot* é uma maneira utilizada para se referir a locais públicos onde disponibiliza acesso à internet, na maioria das vezes através de uma rede sem fio, bastante utilizados em Hotéis, Aeroportos, Universidades entre outros, também utilizados por alguns provedores de serviços de *internet*. A documentação oficial do *RouterOs*, diz que o “conceito de *hotspot* pode ser usado, no entanto, para dar acesso controlado a uma rede, com ou sem fio, através de autenticação baseada em nome usuário e senha” (*Mikrotik*, 2015).

O *routerOs*, possui um firewall poderoso e estável, Alecrin (2013), afirma que firewall são:

Soluções de segurança baseada em *software* e até mesmo em *hardware*, onde a partir de um conjunto de regras ou instruções, analisa o tráfego de dados para determina quais as operações de transmissão ou recepção de dados podem ser executados, o *firewall* é caracterizado como uma “parede” onde por sua vez pode ou não libera o acesso do pacote de dados. O *firewall* atua como uma espécie de barreira onde, verifica quais os dados podem passar ou não, onde essa tarefa só pode ser realizada mediante a regras pré-estabelecidas, e as primeiras soluções de *firewall* surgiram na década de 1990, baseado na filtragem de pacotes de dados.

O *firewall* é normalmente usado como ferramenta de segurança para prevenir o acesso não autorizado a rede interna. Além de segurança no *firewall* são desempenhadas funções como classificação e marcações de pacotes, ainda a classificação de tráfego pode ser baseada em vários classificadores de endereço MAC, endereços IP, Portas, tamanhos de pacotes, entre outros. No *RouterOs* as funções do *firewall* são divididas em espécies de classes, entre elas estão:

- *Filter Rules*: Onde é feita as regras de filtragem de pacotes.
- *NAT⁵*: responsável pela tradução de endereços e portas.
- *Mangle*: Marcação de pacotes conexão e roteamento.
- *Services Ports*: Onde se é possível localizar os *NAT Helpers*.
- *Connections*: onde fica localizado as conexões existentes.
- *Address List*: Lista de endereços IPs inseridos de forma dinâmica ou estática e que podem ser utilizadas em várias partes do *firewall*.
- *Layer 7 protocols*: onde acontece os filtros da camada 7.

O *firewall* opera por meio de regras, onde uma regra é uma expressão lógica que diz para o roteador o que fazer com um tipo particular de pacotes. As regras são organizadas em canais, onde existem três canais padrões.

- *Input* que é responsável pelo tráfego que chega ao *router*.
- *Output* responsável pelo tráfego que sai do *router*.
- *Forward* que é responsável pelo tráfego que passa pelo *router*.

2.3. SISTEMA DE INFORMAÇÃO

Sistema de informação está ligado a qualquer sistema, automatizado ou até mesmo manual, que possam coletar dados e processá-los ou até mesmo organizá-los, de modo que podem envolver pessoas e métodos para coleta de dados. Segundo Mulbert e Ayres (2005),

Sistemas de informação como um conjunto de componentes inter-relacionados, desenvolvidos para coletar, processar, armazenar e distribuir informação, buscando facilitar a coordenação, o controle, a análise, a visualização e o processo decisório, portanto pode-se dizer que trata-se de uma combinação estruturada de práticas de trabalho bem relacionadas as tecnologias de informação, de forma que se consiga um aperfeiçoamento nos objetivos da organização.

⁵ Protocolo que faz a tradução dos endereços IP e portas TCP da rede local para a internet.

Matsuda (2001) afirma que,

Sistemas de Informação são processos administrativos que envolvem processos menores que integram entre si e podem ser divididos em subsistemas menores. De modo que são úteis e coletivos o bastante de para que possa ser aplicado em diversas situações do mundo real.

Um sistema pode ser composto por vários subsistemas interligados. Cada subsistema pode guardar informações de acordo com as funcionalidades da empresa estando interligados com os outros setores, ou seja, subsistemas se relacionam entre si. “Todo sistema, usando ou não sistemas tecnológicos, que manipula e gera informação pode ser genericamente considerado sistema de informação”, diz Rezende (2013).

Para melhor conceituar sistemas de informação, é importante saber diferenciar dado de informação, de modo que o dado consiste como elemento da informação, que capturado isoladamente não apresenta nenhum conhecimento, assim este não apresenta resultados específicos. Ao oposto de dado, informação já trata de elementos mais concisos, onde informação é apresentada como todo dado trabalhado, útil tratado com valor significativo, com sentido lógico para quem usa essa informação.

De acordo com Mulbert e Ayres (2005), um sistema de informação é um produto que gira em torno de três componentes: organizações, pessoas e tecnologia. As organizações podem ser vistas como grandes coleções de processo organizacional e administrativo. As pessoas são os usuários que usam a informação de um sistema para executar seu trabalho. E as tecnologias é o meio pelo qual o sistema de informação pode ser implementado.

Em geral, sistema pode ser classificado como um conjunto de componentes que interagem para atingir determinado objetivo. Já a informação é o conjunto de fatos organizados de modo que produz um valor adicional. Assim um sistema de informação é um tipo especializado de sistema e pode ser definido de diversas formas distintas, por sua vez ainda é um conjunto de elementos ou componentes inter-relacionados que coletam, manipulam e disseminam dados e informação, e oferecem um mecanismo de retroalimentação para atingir um objetivo (STAIR; REYNOLDS, 2006).

Desta forma, existe uma diversidade enorme em conceituar sistemas de informação, assim o mesmo pode ser aplicado em diversas áreas, contudo em cada uma delas os sistemas possuem objetivos e particularidades diferentes. Os sistemas

são importantes para facilitar a execução de um conjunto de ações. Os sistemas estão se tornando cada vez mais usuais e proporcionando ao usuário autonomia, segurança e agilidade no desenvolvimento das suas atividades. Os sistemas estão agregando mais valor quando podem ser acessados de qualquer lugar que o usuário se encontra independente de plataforma ou dispositivo o usuário dispõe, por exemplo, os sistemas *web*.

2.4. SISTEMAS WEB

De acordo com Zaneti Junior (2003), a *Word Wide Web* (WWW) ou simplesmente *web* foi desenvolvida para ser uma fonte do conhecimento humano que permitisse com que vários colaboradores em locais extremamente distintos pudessem compartilhar ideias e todos os aspectos de um projeto comum. O mesmo ainda fala que a *web* deveria permitir que documentos desenvolvidos separadamente pudessem ser juntados facilmente de forma que fossem visualizados sem exigir grandes mudanças e feitas de forma centralizadas.

Zaneti Junior (2003), ainda afirma que as tecnologias *web* podem ser definidas como um sistema de padrões onde inclui:

- **Padrão de endereçamento** os recursos da *web* têm um endereço único e podem ser localizados de qualquer lugar independentemente da plataforma que o mesmo resida.
- **Padrão de comunicação** a tecnologia *web* utiliza um protocolo de comunicação, ou seja, uma linguagem que permita a solução e obtenção de recursos *web*.
- **Padrão de estruturação das informações** o padrão inicial da tecnologia da *web* para apresentação das informações está baseado em uma linguagem de marcação.

Quando existe interação entre usuário e um sistema qualquer dar-se a denominação de aplicação *web*, no entanto muitos utilizam esse termo quando estão se relacionando a sites estáticos, ou seja, onde há somente leitura não interagindo o usuário com o sistema.

As aplicações *web* pode auxiliar bastante os internautas no desenvolvimento de tarefas e até mesmo no gerenciamento de informações. Esse tipo de sistema ajuda na tomada de decisões e controla todo fluxo de informação que ocorre dentro das

organizações. Outra característica importante também é a capacidade que seus usuários têm de acessá-lo, independente de que plataforma estejam e onde estão localizados.

Para execução do projeto foi necessário à utilização de mecanismos importantes para desenvolvimento das atividades previstas, assim os usos de tecnologias são indispensáveis para conclusão das aplicações, sendo necessário o conhecimento técnico e específico no desenrolar da aplicação *web*. Existem diversas maneiras de atingirmos o fim principal deste estudo, dentre os quais podemos destacar:

2.5. TECNOLOGIA DE INFORMAÇÃO

A tecnologia de informação (TI) tem evoluído bastante, com isso surgem cada vez mais soluções e tendências tecnológicas utilizadas para atingir determinado objetivo específico. Além disso, manipulam e coordenam informações, patrimônio valioso das organizações, de modo que faz-se necessário utilizar ferramentas que proporcionam bons resultados com o menor custo benefício.

Alecrin (2011) define TI como o conjunto de todas as atividades e soluções providas por recursos computacionais que visam permitir a obtenção, o armazenamento, o acesso, o gerenciamento e o uso das informações. Dispõem ainda que as aplicações destas estão ligadas nas mais diversas áreas, de modo que não há uma definição que as expressa por completo.

Além disso, Cruz (2009) fala que tecnologia de informação pode ser todo e qualquer dispositivo capaz de tratar dados ou informação, tanto de forma sistêmica como de forma ocasional, quer esteja aplicada ao produto ou até mesmo ao processo.

Como complemento destes conceitos, podem ser apresentados alguns componentes onde se fundamenta as tecnologias de informação:

- *Hardware* e seus dispositivos periféricos.
- *Software* e seus recursos.
- Sistemas de telecomunicação.
- Gestão de dados e informação.

De acordo com os conceitos e definições dos autores pode-se concluir que, o uso de tecnologias de informação consegue melhorar a comunicação e a organização dos dados, em vários setores das organizações, como por exemplo em

escolas, órgãos públicos, empresas entre diversos outros campos, além disso pode-se dizer que o uso das tecnologias contribui para estender as iniciativas públicas e privadas que com o passar do tempo pode tornar-se um instrumento de predominância no ramo tecnológico.

2.6. LINGUAGENS DE PROGRAMAÇÃO

Antes de abordarmos o que vem a ser linguagem de programação vamos entender um pouco do conceito de algoritmos. De acordo com Xavier (2010), um algoritmo é caracterizado por qualquer forma de resolver problemas a partir de padrões e regras. O algoritmo computacional, nada mais é do que um passo a passo que viabiliza o processo de resolver determinada tarefa.

Ferrari e Cechinel (2008) fala que um algoritmo pode ser definido como uma sequência finita de passos para resolver determinado problema, assim ao desenvolver um algoritmo é estabelecido padrões de comportamentos que deverá ser seguido para alcançar o resultado.

Ao definir linguagem de programação Ferrari e Cechinel (2008), dispõe que qualquer tipo de informação que deva ser transferida, processada ou armazenada deve estar na forma de uma linguagem, sendo imprescindível para o processo de comunicação.

Linguagem natural é aquela utilizada pelos seres humanos para troca de informações e expressar seu raciocínio de forma a serem compreendidos pelos demais. Já os computadores foram projetados para executarem tarefas que lhes foram determinadas a partir de instruções pré-estabelecidas, ou seja, a linguagem natural não é compreensível pelas máquinas, uma vez que estas somente abstraem as instruções escritas em sua linguagem habitual, composta por números e representada na forma binária.

As linguagens de programação podem ser classificadas em dois níveis: as de baixo nível assemelhadas a linguagem da máquina; e as de alto nível que se aproximam a linguagem natural, os exemplos mais comuns dessas linguagens estão *Pascal*, *C*, *Fortran*, *Java*, *Perl*, *Lisp*, *PHP*. Assim como o *Python* que é a linguagem utilizada pelo projeto em tese.

2.7. PYTHON

Python é uma linguagem de alto nível, orientada a objeto de tipagem forte e dinâmica, interpretada e interativa. Venners (2003) fala a respeito do Python:

O *Python* possui uma sintaxe objetiva onde ajuda o programado se aperfeiçoar e se integrar cada vez mais com a linguagem, assim aumentando a produtividade e facilitando no desenvolvimento dos códigos por parte do programador.

Segundo Borges (2010), a linguagem *Python* possui uma estrutura de alto nível de modo que apresenta uma grande coleção de módulos prontos para ser usada. A história do Python é descrita da seguinte forma:

Python foi criado em 1990 pelo físico Guido Van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI), com foco principal em usuários como físicos e engenheiros. Atualmente a linguagem aceita por grandes empresas de tecnologias como: *Google, Yahoo, Microsoft, Nokia, Disney* (DORGES, 2010).

Python é uma linguagem de código aberto com licença compatível com a GPL (*General Public License*), no entanto menos restritiva, permitindo que seja incorporado a produtos proprietários, podendo ainda ser utilizado como linguagem principal para desenvolvimento de sistemas.

Alguns autores tratam *Python* como uma linguagem de *scripting*, porem Reis (2010), discorda ao afirmar é de uso geral, uma vez que pode ser utilizado para criar qualquer tipo de *software*. O termo *script* geralmente é usado para programas escritos em linguagem interpretada que automatizam tarefas ou que se conecta a programas distintos. O autor ainda classifica alguns pontos onde faz com que o *Python* seja tão especial:

- É uma linguagem interpretada.
- No *Python* não há pré-declaração de variáveis, e os tipos de variáveis são declaradas dinamicamente.
- O controle de bloco é feito apenas por endentação.
- Oferece tipos de alto níveis: *string*, lista, *tuplas* arquivos, etc.
- É orientada a objetos: Alias, em *Python* tudo é objeto.

Baseado nos conceitos abordados pelos autores supracitados, percebe-se que a linguagem *Python* é propensa para elaboração do projeto, sendo que a mesma desfruta de uma linguagem usual e de fácil aprendizado, livre e multiplataforma. Pode-

se resumir que o *Python* é uma linguagem expressiva que permite fácil implementação dos algoritmos, além do mais possui uma comunidade ativa fabulosa e seus usuários estão dispostos a contribuir para solução de problemas.

A biblioteca *Paramiko* em *Python* trata-se da implementação do protocolo SSHv2, fornecendo funcionalidades cliente e servidor, enquanto aproveita uma extensão *Python*. A própria *Paramiko* é uma *interface Python* puro no que diz respeito aos conceitos de rede SSH, este módulo porém, pode ser instalados com outros módulos *Python*, apesar do *PyCrypto* ser escrito grande parte em C. *Paramiko* em si possui uma vasta documentação da API e uma lista de discussão ativa, o mesmo também oferece uma implementação do SSH e protocolos de servidor de SFTP.

Quando utilizamos *Paramiko* em *Python*, fazemos uso do protocolo de comunicação SSH, protocolo esse que permite a execução de comandos em unidades remotas através scripts enviados por meio da rede.

Dentro das Tecnologias *web* trabalha-se com um protocolo de rede chamado *Real-time Transport protocol (RTP)* que é utilizado para aplicações em tempo real na entrega de áudio ponto-a-ponto como por exemplo *Voice over Internet Protocol (VOIP)*, nele fazendo a fragmentação do fluxo de dados, adicionado a cada fragmento informação e tempo de entrega. O controle de fluxo é feito pelo *Real-Time Control Protocol (RTCP)*, e ambos utilizam UDP como protocolo de transporte, o qual não dar nenhuma garantia que os pacotes sejam entregues num determinado intervalo de tempo (PERKINS,2003).

Atualizações em tempo real na *web* se tornaram um aspecto importante, com surgimento de aplicações como bate papo, jogos *online*, redes sociais, vídeo chamadas, VOIP e em inúmeras outras aplicações. No *Python* a aplicação de real time é feita através de uma biblioteca chamada *Tornado*.

De acordo com Recordon (2009), o *Tornado* é uma estrutura relativamente simples de não bloqueio em servidor *web*, escrito em *Python* e projetado para lidar com milhares de conexões simultâneas, tornando-o ideal para serviços *web* em tempo real. O mesmo autor ainda afirma que o *tornado* é uma peça fundamental da infraestrutura, com poderes e funcionalidades em tempo real, pois o mesmo se destaca com a velocidade e a manipulação de grande quantidade de trafego simultâneo.

O *Tornado* é um *framework* desenvolvido em *Python* com uma biblioteca de rede assíncrona originalmente desenvolvido para uso em rede de não bloqueio de entrada/ saída, podendo escalar dezenas de milhares de conexões simultâneas, ideal

para aplicações como *WebSockets* e outras aplicações que exigem conexões de longa durações para cada usuário.

2.8. DJANGO

Para podermos utilizar todos os recursos que o *Python* oferece é necessário ter conhecimento sobre o conceito de *framework* e sobre quais vantagens essa ferramenta pode trazer no decorrer do projeto.

Podemos dizer que um *framework* é uma ferramenta que auxilia no desenvolvimento garantindo assim uma maior reutilização de códigos, auxiliando na agilidade do processo, podendo ainda pode ser representada como um conjunto de classes abstratas e concretas, com grande potencial de especialização.

A Documentação Oficial (2009) afirma que *Django* é um *framework web* de alto nível, construído integralmente sobre a linguagem de programação *Python*, com o objetivo principal de estímulo para o desenvolvimento de sistemas rápidos e limpos, reduzindo tempo e permite construir aplicações *Web* de qualidade e fácil aplicação. A definição de Django é apresentado a seguir:

O *Django* utiliza o ORM (Mapeamento Objeto-Relacional) que, basicamente, transforma objetos em estados num banco de dados. A camada de controle é comumente implementada utilizando-se da estrutura, que é responsável pelas regras de negócio. Na camada de visão é utilizada a própria estrutura, à respeito da renderização dos *templates*, para a configuração das *urls* e os *templates* propriamente ditos. (FONSECA e BRAGA, 2009)

Um das funções que chama bastante atenção no *Django* é a criação automática de interfaces administrativas simples e amigáveis, de modo que auxilia o programador a realizar tarefas simples que em muitos casos seriam repetitivas. O motivo pelo qual foi escolhido o *framework Django*, foi a familiarização pessoal com a ferramenta.

2.9. ENGENHARIA DE SOFTWARE

De acordo com Sommerville (2003), a engenharia de *software* baseia-se numa disciplina que tem por foco todos os aspectos da produção de *software*, que vai desde os estágios iniciais das especificações do sistema até sua manutenção mesmo quando em uso. Dizemos que a engenharia não se preocupa apenas com o processo

técnico, mas sim com todas as fases de elaboração. Além do mais é através da mesma que se destaca: o processo de gerenciamento de projeto, desenvolvimento de ferramentas, métodos e teorias para apoio a produção de *software*.

Conforme Pressman (2011), o processo de *software* possibilita o desenvolvimento de atividades e escolha do conjunto adequado de ações e tarefas. O autor ainda estabelece um alicerce para as etapas de engenharia complexa que engloba um conjunto de atividades de apoio aplicável em todo processo de *software*. Essa atividade baseia-se em:

- Comunicação: antes que se inicie qualquer trabalho técnico é importante a comunicação com os *Stakeholders*.
- Planejamento: deve existir plano de projeto, onde descreve-se as tarefas técnicas a serem executadas.
- Modelagem: é necessário elaborar um esboço para ter em mente a visão do projeto em geral.
- Construção: atividade na qual combina geração de código e testes.
- Emprego: momento em que o *software* é entregue ao cliente podendo testá-lo e depois fornece um *feedback*.

Filho (2003) conceitua processo como um conjunto de fases ordenadas, compostas por atividades, métodos, práticas e transformações usadas para se auferir metas idealizadas, estando estas por diversas vezes ligada a um ou mais resultados finais que compõe o produto da execução do processo.

Diante do exposto dizemos que durante as fases de elaboração do projeto de *software*, é de suma importância a execução de tarefas que aumentem a qualidade do produto, fornecer aos membros soluções ágeis que ampliam a produtividade da equipe preservando de tal forma a satisfação do cliente.

Para desenvolver um *software* passamos por diversas fases que compõe sua elaboração, definido atividades, funções e responsabilidades dos membros da equipe. Cabe destacar que cada *software* possui etapas distintas, uma vez que detêm características e peculiaridade diversas, o que não exclui a possibilidade serem adotados os mesmos procedimentos em novos projetos.

Os modelos de ciclo de vida são apresentados como estruturas pré-definidas que se adequa as fases do processo. Estão presentes no ciclo de vida: processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção no

produto de *software*, envolvendo toda a vida do sistema, desde as definições de seus requisitos até ao termino e uso do produto (MACÊDO e SPÍNOLA, 2011).

A Figura 1 representa os seis estágios do ciclo de vida da aplicação.

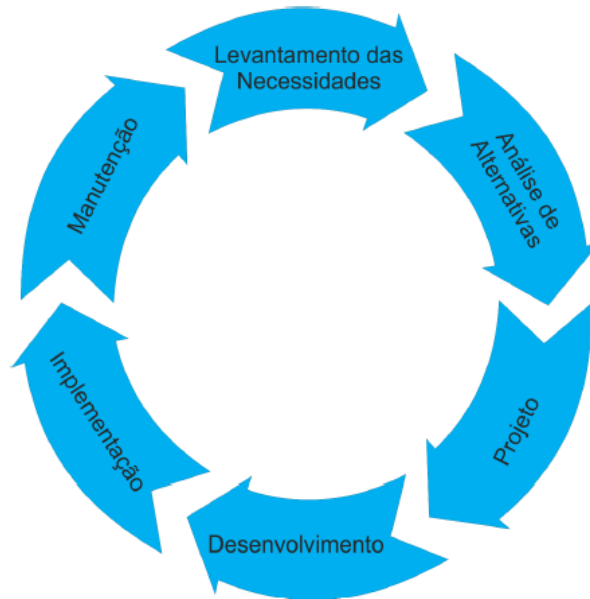


Figura 1 - Ciclo de vida da aplicação
Fonte Gordon e Gordon (2006)

Ao tratar do ciclo de vida do *software* Gordon e Gordon (2006) relata que este está relacionado com os estágios de concepção, projeto, criação e implementação de um sistema de informação.

Etapas que compõe o ciclo de vida da aplicação:

- Levantamento de Necessidades: Etapa de coleta de dados sobre as necessidades dos usuários;
- Análise de alternativas: Definição das tecnologias a serem utilizadas.
- Projeto: Especificações detalha do sistema proposto;
- Desenvolvimento: criação ou aquisições de *software* necessário para implementação do projeto;
- Implementação: processo de ativação de um novo sistema;
- Manutenção: etapa onde ocorre a correção de erro ou modificação do sistema;

O planejamento adequado de cada ciclo pelo qual o *software* se submete será essencial para que se logre êxito, estabelecendo os caminhos a serem percorridos que vai desde o levantamento dos requisitos necessário a elaboração até depois de sua conclusão. Ao aderir esta postura podemos controlar o desenvolvimento da maneira mais eficaz além de explorar todo o procedimento pelo qual o *software* segue.

3. ETAPAS E FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO DO SCG-MK- SISTEMA DE GERENCIAMENTO E CONTROLE DE SERVIDORES DA MIKROTIK.

O objetivo geral deste projeto é o desenvolvimento do protótipo de uma aplicação *web*, para gerenciamento e controle de servidores da *Mikrotik*. Portanto neste capítulo será abordado as principais técnicas de produção de *software* e seu desenvolvimento ágil, ilustraremos ainda as funcionalidades do sistema SCG-MK, de forma a detalhar as principais telas do sistema.

Assim para o desenvolvimento do protótipo funcional, foi utilizada a linguagem de programação *Python*, o motivo da escolha está relacionado a suas características que a torna simples, ou seja, de fácil aprendizado, que se exterioriza de forma clara e

objetiva proporcionando facilidades em manutenções futuras, além de ser modular permitindo a escalabilidade da aplicação independentemente da situação.

As diferentes ferramentas que auxiliam na elaboração de sistemas, geralmente utilizam padrões de projetos definidos, que podem automatizar diversas tarefas repetitivas, além de proporcionar ao programador uma menor complexidade em possíveis problemas que podem surgir no decorrer do projeto.

Uma das ferramentas utilizada foi o *framework Django*, esta trata-se de um *framework* escrito integralmente em *Python*, possuindo recursos atraentes que aumentam a produtividade do desenvolvedor, além de dispor de bibliotecas e documentação que contém todos os elementos necessários para o entendimento da ferramenta e suas funções.

Para a elaboração do projeto, foi necessário a realização de uma análise de viabilidade e aplicabilidade do sistema e em seguida houve o levantamento de requisitos do sistema. Essa etapa de execução do projeto foi importante para o correto desenvolvimento a partir das necessidades levantadas. No Quadro 1 são mostrados os requisitos funcionais do sistema, identificados por um código único, com suas respectivas descrições e dependências em casos específicos.

Quadro 1 - Requisitos Funcionais

IDENTIFICADOR	DESCRIÇÃO	DEPENDENTE DE
RF01	O sistema deverá ter um usuário administrador por padrão	
RF02	O sistema deve permiti o gerenciamento de usuário	RF01
RF03	O sistema deverá fornecer autenticação através de <i>login</i> .	RF02
RF04	O sistema possibilita o gerenciamento de alguns itens: servidores, planos, fluxo de caixa, clientes.	RF03
RF05	O sistema deverá permiti o gerenciamento de vários servidores	RF04
RF06	O sistema deverá permiti, adicionar, edita remover os servidores	RF04
RF07	O usuário poderá adicionar, remover, alterar, listar os clientes	RF04
RF08	O usuário pode adicionar, remover, alterar e listar planos	RF04

RF09	O sistema deverá permitir que o usuário cadastre, altere os pagamento de títulos	RF04
RF10	O sistema deverá emitir relatórios de caixa, listar cobrança em aberto, listar faturamento mensal.	RF04
RF11	O sistema deverá monitorar a saída e entrada de link por interface e uso de memória e CPU	

No Quadro 2 são apresentados os requisitos não funcionais do sistema identificados por um código único, seguido de suas descrições e respectiva categoria pela qual o requisito se enquadra.

Quadro 2 - Requisitos não Funcionais

IDENTIFICADOR	DESCRIÇÃO	CATEGORIA
RNF01	O sistema deve ser implementado usando a Linguagem de Programação Python	Implementação de padrões
RNF02	O sistema deve apresenta telas intuitivas e de fácil aprendizado	Facilidade
RNF03	O sistema deve ser acessível independente de sua plataforma	Padrões
RNF04	O sistema deve ser implementado usando técnicas de responsividade.	Portabilidade
RNF05	O sistema deve processar milhares de requisição por segundo	Eficiência
RNF06	O sistema deve estar disponível na maior parte do tempo	Disponibilidade

Após analisar os requisitos, foram construídos os diagramas das funcionalidades da aplicação utilizando os conceitos descritos pela UML. De acordo com Guedes (2011), a UML surgiu da união de três métodos de modelagem: do *Booch*, o OMT (*Object-Modeling Technique*) e o OOSE (*Object-Oriented Software Engineering*),

A UML é uma linguagem utilizada para modelar *software* que utilizam o paradigma de orientação a objetos e é considerada como uma linguagem de uso geral podendo ser utilizada em todos os domínios da aplicação.

Pressman (2011), fala que a UML é uma linguagem padrão usada para descrever, visualizar, especificar, construir e documentar projetos de *software*, de

modo a facilitar o entendimento do sistema, tanto para quem está desenvolvendo como para os demais *Stakeholders*⁶.

A UML é composta por diversas espécies de diagramas com o objetivo de fornecer melhor imagem do sistema a ser construído, o diagrama analisará o sistema ou parte dele.

Exemplos de diagramas UML:

- Diagramas de Casos de Uso: é considerado como um diagrama geral e informal da UML, geralmente utilizado no levantamento de requisitos e análises de sistema.
- Diagrama de Classe: é um dos mais importantes diagramas que serve de apoio para outros. Sua finalidade é justamente definir a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos, além de estabelecer como estas se relacionam e trocam informações entre si.
- Diagrama de Objeto: é um complemento ao diagrama de classes e dependente deste, fornecendo uma visão dos valores armazenados pelos objetos de um diagrama de classes em determinado momento da execução do processo do *software*.
- Diagrama de Sequência: é um diagrama comportamental que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em determinado processo.

3.1. PROCESSO DE CONFIGURAÇÃO DO *ROUTEROS*

No processo de configuração normal do *RouterOs*, são difundidas de duas maneiras principais. A primeira através do terminal, nele os comandos são inseridos

⁶ Todas as partes interessadas no projeto.

através de linhas de códigos, como é mostrado na Figura 2 (a).

```

MMM      MMM      KKK
MMMM     MMMM     KKK
MMM MMMM  MMM  III  KKK  KKK  RRRRRR  000000  TTT  III  KKK  KKK
MMM  MM  MMM  III  KKKKK  RRR  RRR  000 000  TTT  III  KKKKK
MMM      MMM  III  KKK  KKK  RRRRRR  000 000  TTT  III  KKK  KKK
MMM      MMM  III  KKK  KKK  RRR  RRR  000000  TTT  III  KKK  KKK

MikroTik RouterOS 5.20 (c) 1999-2012      http://www.mikrotik.com/

[admin@MikroTik] >

```

Figura 2 - Terminal do RouterOs

A segunda maneira é através da ferramenta Winbox, como é mostrado na Figura 3.

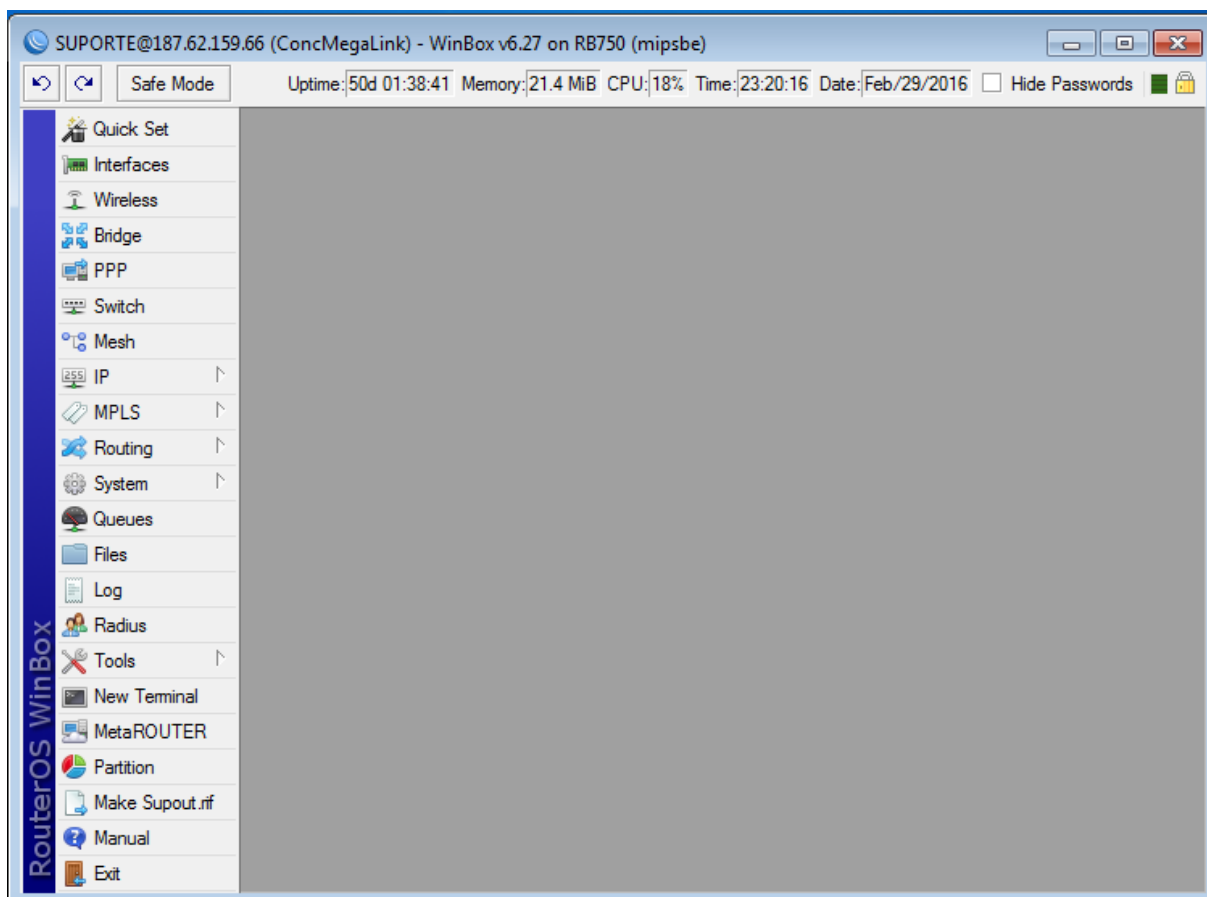


Figura 3 - Ferramenta de Winbox

As configurações via terminal são realizadas através de scripts, porém no Winbox são feitas através de marcações, como são mostrados na Figura 4 e 5 que mostra como é realizado um redirecionamento dentro do *RouterOS*.

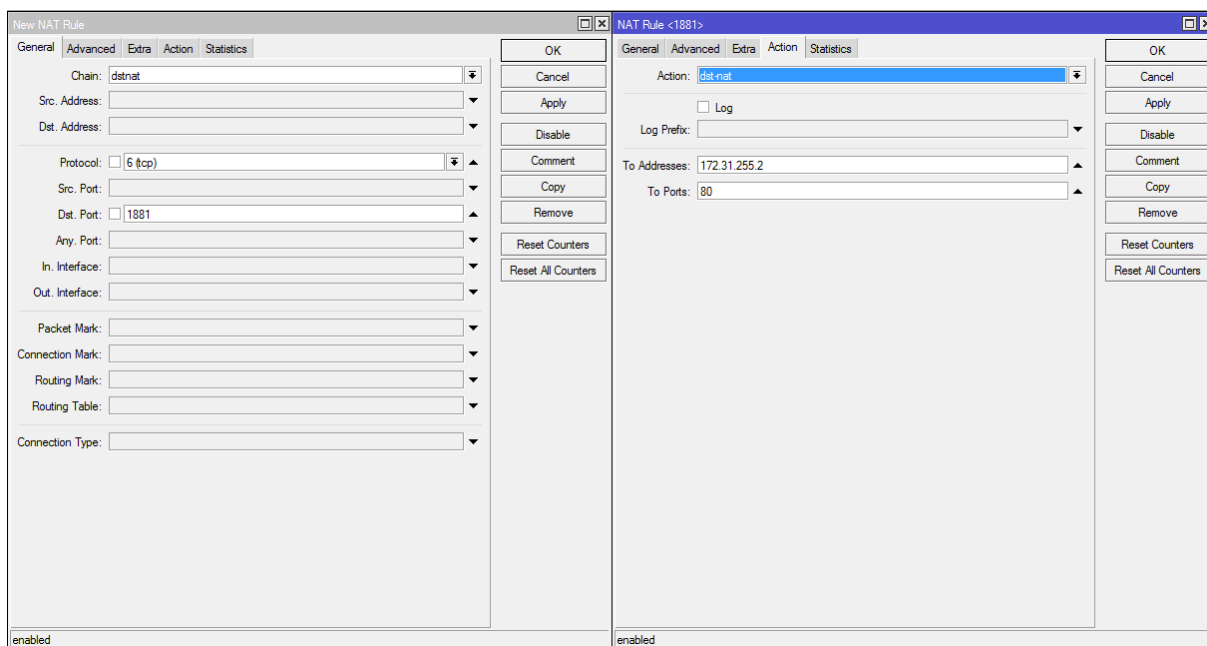


Figura 4 - Configuração de redirecionamento pelo WinBox

```

MMMM   MMMM   KKK           TTTTTTTTTT   KKK
MMM MMMM MMM III  KKK KKK RRRRRR   OOOOOO   TTT   III  KKK KKK
MMM MM  MMM III  KKKKKK  RRR RRR  OOO OOO   TTT   III  KKKKK
MMM   MMM III  KKK KKK  RRRRRR   OOO OOO   TTT   III  KKK KKK
MMM   MMM III  KKK KKK  RRR RRR   OOOOOO   TTT   III  KKK KKK

MikroTik RouterOS 6.27 (c) 1999-2015      http://www.mikrotik.com/

[?]          Gives the list of available commands
command [?]  Gives help on the command and list of arguments

[Tab]       Completes the command/word. If the input is ambiguous,
            a second [Tab] gives possible options

/           Move up to base level
..         Move up one level
/command    Use command at the base level

[SUPORTE@ConcMegaLink] > ip firewall nat add action=dst-nat chain=dstnat comment=
[SUPORTE@ConcMegaLink] > ip firewall nat add action=dst-nat chain=dstnat comment="RE
DIRECIONAMENTO MK-AUTH" disabled=yes dst-port=89 protocol=tcp to-addresses=172.31.25
5.2 to-ports=89

```

Figura 5 - Redirecionamento via Terminal

Os exemplos abordados anteriormente servem como base para ter conhecimento de como são feitas as configurações manuais no *RouterOs*, a seguir serão apresentados como são realizadas as configurações através do sistema desenvolvido.

3.2. ENTENDENDO O SISTEMA

A Figura 6 ilustra o diagrama de casos de uso referente às principais funcionalidades do sistema. Observe que existem dois tipos de usuários que podem operar o sistema: o primeiro usuário denominado suporte, irá gerenciar o sistema com o privilégio de administrador, o mesmo poderá cadastrar servidores e planos, gerenciar servidor de DHCP, além de alterar as configurações do *firewall*. O segundo usuário denominado atendente possui menos privilégios, o mesmo poderá gerenciar os clientes, ou seja, conta com a tarefa de cadastrar, edita e excluir clientes, além de fazer o gerenciamento do caixa, no cadastramento e recebimento de cobranças e emitir relatórios.

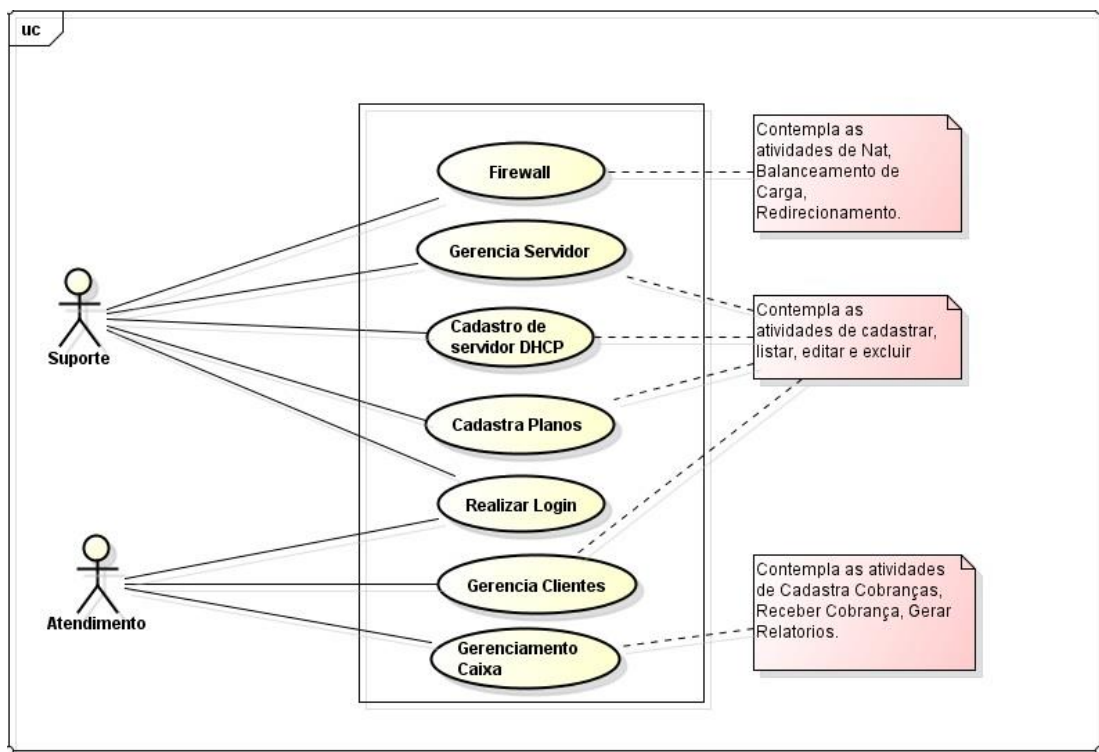


Figura 6 - Diagrama de Caso de Uso

A Figura 7 mostra o diagrama de classes do subsistema de autenticação. Onde a classe "Usuário" representa os usuários do sistema com seus atributos, essa classe se relaciona com a classe "Permissão", que representa as permissões do sistema, cada usuário pode ter uma ou várias permissões e uma mesma permissão pode ser dada a mais de um usuário. A classe "Usuário" se relaciona também com a classe "Grupos" que representam um conjunto de permissões, formando assim grupos de usuários, para isso a classe "Grupos" também se relaciona com a classe

“Permissão”. A classe “Log” representa o *log* de operações realizadas pelos usuários, por isso a classe também se relaciona com “Usuário”.

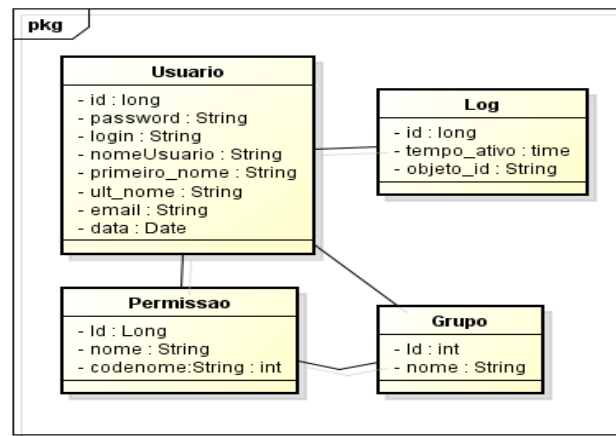


Figura 7 - Diagrama de Classe Autenticação

Na Figura 8 é apresentado o diagrama de classes: a classe “Servidor” é responsável por fazer o gerenciamento dos servidores cadastrados, a classe “Interface” que se relaciona com “Servidor” guarda as informações sobre as *interfaces* cadastrada, as classes “Servidor_DHCP”, “Plano”, “Interface_Nat” estão relacionadas a “Interface”, logo “Cliente” está ligado a “Plano”.

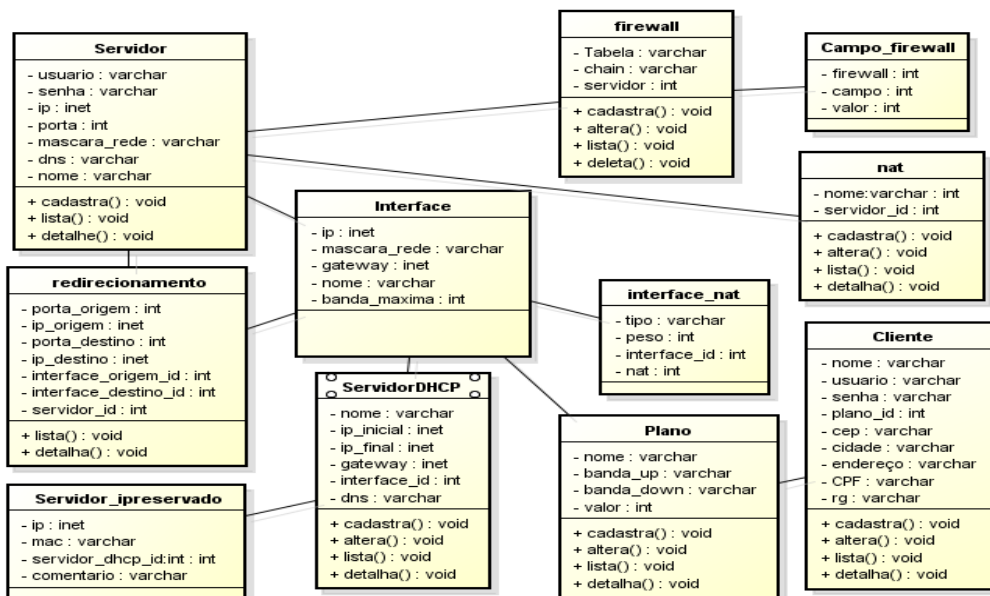


Figura 8 - Diagrama de Classe

3.3. TELA DE LOGIN

Na tela de *login* ou tela de autenticação, Figura 9, o usuário do sistema deverá autenticar-se com usuário e senha, como foi mostra no diagrama de caso de uso na Figura 6, o sistema é capaz de gerenciar dois tipos de usuários, o primeiro que é o

usuário suporte onde o mesmo possui privilégios de administrador do sistema, e o segundo responsável pela parte financeira.

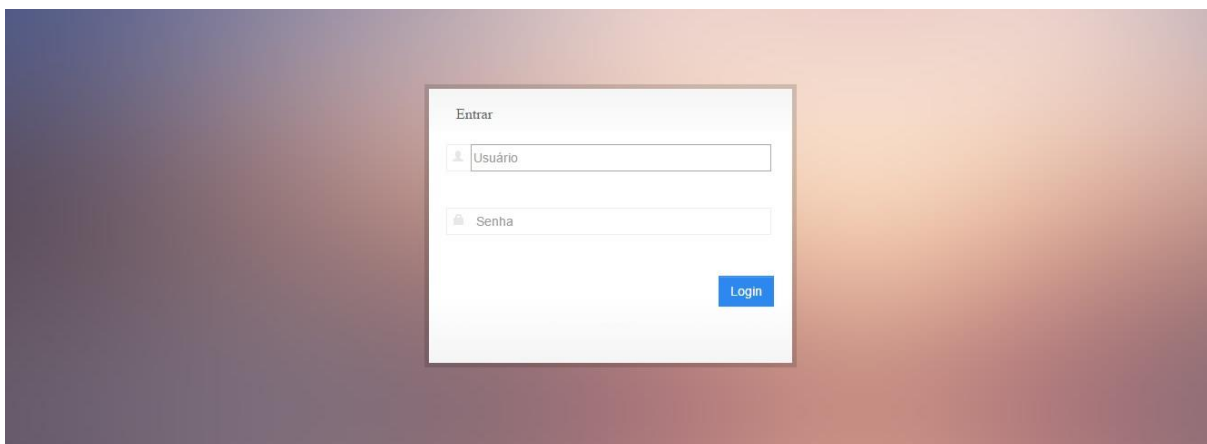


Figura 9 - Tela de Autenticação

3.4. TELA INICIAL

A Figura 10 apresenta a tela inicial do sistema, cuja função é exibir ao usuário o tráfego de dados que está sendo transferido por cada *interface* do servidor, o uso de memória e quantidade de clientes conectados.



Figura 10 – Tela Inicial

3.5. CADASTRO DE SERVIDORES

A Figura 11 ilustra a tela de cadastro dos servidores. Para o cadastro, é necessário informar, o nome do servidor, IP de acesso e a porta na qual o servidor se conecta, lembrando que a porta padrão será a 22, podendo a mesma ser alterada e logo em seguida informa-se usuário e senha do respectivo servidor.

SCG-MK. Sistema de Configuração e Controle de Servidores da Mikrotik

Home > Dashboard

Cadastro Servidor

Nome

Ip

Porta

Usuario

Senha

Salvar

© 2013 Bootstrap Metro Dashboard

Figura 11 - Cadastro de Servidores

3.6. LISTA DE SERVIDORES

O sistema é típico para gerenciar e configurar vários servidores simultaneamente. Assim, a Figura 12 mostra a lista de todos os servidores cadastrados no sistema. Existe a possibilidade de o usuário realizar algumas modificações nos servidores cadastrados, como visualizar detalhes, editar configurações ou até mesmo excluir um servidor e fazer ainda configurações no *Firewall*, ou seja, regras de redirecionamento e NAT.

SCG-MK. Sistema de Configuração e Controle de Servidores da Mikrotik

Home > Dashboard

Lista de Servidores

Nome	IP	Usuario	Porta	Ações
teste	192.168.1.22	admin	22	    
teste	192.168.2.1	admin	22	    
mk	10.180.70.9	admin	24	    

© 2013 Bootstrap Metro Dashboard

Figura 12 - Lista de Servidores

3.7. CADASTRO DE PLANOS

A Figura 9 mostra como o usuário realiza o cadastro dos planos de acesso, o mesmo deverá informar o nome do plano, o valor de *Upload* e *Download* e o valor desse plano.

SCG-MK. Sistema de Configuração e Controle de Servidores da Mikrotik

Home > Dashboard

Cadastro Planos

Interface: mklink

Nome:

Banda up:

Banda down:

Valor:

Salvar

Figura 13 - Cadastro de Planos

3.8. CADASTRO DE CLIENTES

A Figura 14 ilustra como procede o cadastro de clientes, aqui o usuário deverá preencher todos os campos referentes aos dados pessoais do cliente, além de selecionar qual o plano e interface o mesmo irá se conectar.

SCG-MK. Sistema de Configuração e Controle de Servidores da Mikrotik

Home > Dashboard

Cadastro Clientes

Nome:

Usuario:

Senha:

Servidor: teste
teste
mik

Plano:

Cpf:

Rg:

Endereço:

Cep:

Cidade:

Salvar

Figura 14 - Cadastro de Clientes

3.9. TELA DE PAGAMENTO

Na Figura 15 apresentamos a tela de pagamento, nesta tela é emitido relatórios a respeito de cobranças e pagamentos, é possível ainda receber cobrança, excluir ou alterar data de pagamento.

Franciso jose da silva				
Mes	Ano	Data Vencimento	Pago	Ações
01	2016	03/01/2016	Pago	  
02	2016	03/02/2016	Pago	  
03	2016	03/03/2016	Aguardando	  
04	2016	03/04/2016	Aguardando	  
05	2016	03/05/2016	Aguardando	  
06	2016	03/06/2016	Aguardando	  
07	2016	03/07/2016	Aguardando	  

Figura 15 - Tela de Pagamento

4. CONSIDERAÇÕES FINAIS

Este projeto foi desenvolvido com a finalidade de auxiliar os gerentes de rede a ter melhor controle sobre os equipamentos que trabalham com o *RouterOs*, além de proporcionar automatização nas configurações no sistema operacional, de modo que as configurações são enviadas para o servidor de forma despercebida pelo usuário.

O sistema foi desenvolvido para operar em plataforma *web*, com isso foi utilizando técnicas de responsividade, que possibilita aos usuários utilizar o sistema em qualquer dispositivo que tenha acesso a *internet*, com telas adaptáveis e uma navegabilidade agradável.

O projeto encontra-se como protótipo experimental totalmente funcional, apresentando os testes realizados das suas funcionalidades e é através das imagens de tela que podemos compreender melhor o funcionamento do sistema.

O SCG-MK visa agilizar e facilitar o trabalho dos gerentes de rede nas configurações do *RouterOs*, levando ainda um modulo para que se possa monitora o uso dos recursos disponível no servidor. O sistema conta com um gerenciamento financeiro e possibilita a configuração de gerenciamento dos servidores, que auxilia os pequenos provedores no gerenciamento de suas atividades.

Para projetos futuros pretende-se agregar outros recursos ao sistema, entre estes podemos destacar, a implantação de recursos de monitoramento de incidentes, monitorar o uso da internet por cada servidor, com isso o gerente de rede poderá obter informações detalhadas a respeito do uso de carga de cada servidor, além disso será adicionado novos módulos de cobranças.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- ALECRIN, Emerson. **O que é firewall?** : Conceito, tipos e arquiteturas. 2013. Disponível em: <<http://www.infowester.com/firewall.php>.> Acesso em: 22/01/2016
- ALECRIN, Emerson. **O que é tecnologia da informação (TI)?**. 2011. Disponível em: <<http://www.infowester.com/ti.php>.> Acesso em: 05/01/2016
- BORGES, Luiz Carlos; **Python para Desenvolvedores**. Rio de Janeiro: Edição do Autor, 2010.
- BRANDT, Robert Rice. **Sistema Operacional de Redes**. 2006. Disponível em: <<http://rrbrandt.dee.ufcg.edu.br/br/docs/redes/sor>.> Acesso em: 10 de dezembro de 2015
- CASTTELS, ; MANUEL, . **A galaxia da internet**: Reflexão sobre a internet. Rio De Janeiro: Zahar, 2003. p. 7. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=nCKFFmWOnNYC&oi=fnd&pg=PA5&dq=internet+&ots=_CDPZGw82L&sig=-ISPvpafzKI_kbR-xUWbbjZx63Y#v=onepage&q&f=false.> Acesso em: 03/01/2016
- CRUZ, Tadeu. **Sistemas de Informações Gerenciais**. Atlas, São Paulo, Brasil, 2009.
- FERRARI, Fabricio; CECHINEL, Cristian. **Introdução A Algoritmos E Programação**. 2008. Disponível em: <<http://www.ferrari.pro.br/home/documents/documents.php>.> Acesso em: 11 de janeiro de 2015
- FILHO, Pio Armando Benine. **Informática: Conceitos e Aplicações**. Érica, São Paulo, Brasil, 2012.
- FILHO, Wilson de Pádua Paula. **Engenharia de Software**: Fundamentos, Métodos e Padrões. LTC, ro, Brasil, 2003.
- FONSECA, Vinicius Prado da, BRAGA, Felipe Mendes. **Django, Desenvolvimento Ágil para Web**: UFT, 2009.
- GORDON, Steven R.; GORDON, Judith R. **Sistemas de Informação**: Uma Abordagem Gerencial. GEN, Rio de Janeiro, Brasil, 2006.
- GUEDES, Gilleanes T. A. **UML 2: Uma Abordagem Prática**. Novatec, São Paulo, Brasil, 2011.
- JESSE, . **Ssh Programming With Paramiko Completely Different**. 2009. Disponível em:< <http://jessenoller.com/blog/2009/02/05/ssh-programming-with-paramiko-completely-different>.> Acesso em: 10 de janeiro de 2016
- LABAKI, J. **Introdução a Python – Módulo A**: Bem-Vindo a Python, Ilha Solteira: Unesp, s.d. 113 p.

MACÊDO, Ana Bárbara Lins de; SPÍNOLA, Rodrigo. **Ciclos de Vida do Software: Conhecendo os Bastidores**. 2011. Disponível em: <<https://profareane.files.wordpress.com/2013/02/ciclo-de-vida-do-software.pdf>> Acesso em: 13 de janeiro de 2016

MATSUDA. **Teoria Geral dos sistemas**. Disponível em: < <https://pt.scribd.com/doc/72496302/Teoria-Geral-de-Sistemas-Matsuda>>. Acesso em: 05 de Junho de 2015.

MAZZA, Lucas. **HTML5 e CSS3: Domine a web do futuro**. Casa do Código, São Paulo, Brasil, 2012.

MENDES, Douglas Rocha. **Redes de Computadores: Teoria e prática**. Novatec, São Paulo, Brasil, 2007.

Mikrotik. 2015. Disponível em: <http://wiki.mikrotik.com/wiki/Manual:TOC>. Acesso em: 12/01/2015

MIRANDA, Anibal D. A.. **Introdução Às Redes De Computadores**. 2007. Disponível em: <https://fasul.edu.br/portal/files/biblioteca_virtual/7/introducaoaredesdecomputadores.pdf> Acesso em: 24 de dezembro de 2015

MORIMOTO, Carlos E.. **Redes e Servidores Linux : Guia Pratico**. 448. ed. Brasil: GDH Press e Sul Editores, 2006.

MULBERT, Ana Luísa; AYRES, Nilce Miranda. **Fundamentos para Sistemas de Informação**. Palhoça UnisulVirtual, 2005.

PAULI, Josh. **Introdução ao Web Hacking: Ferramentas e técnicas para invasão de aplicações web**. Novatec, São Paulo, Brasil, 2013.

PERKINS, Coli. **RTP : Audio and video for the Internet**. Boston: Pearson Education, 2003. Disponível em: <https://books.google.com.br/books?id=OM7YJAY9_m8C&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acesso em: 10 de janeiro de 2016

PINHEIRO, José Mauricio Santos. **Princípios dos Sistemas Operacionais de Rede**. 2008. Disponível em: <http://www.projetoderedes.com.br/artigos/artigo_principios_sistemas_operacionais_de_rede.php> Acesso em: 04 de janeiro de 2016

PINTO, S. C. C. S.. **Composição em Web Frameworks**, tese de doutorado, Departamento de Informática PUC-Rio, 2000.

PNDA: De 2005 para 2011, o numero de internautas cresce 143,8%. IBGE(Instituto Brasileiro de Geografia e Estatística); Disponível em: <<http://saladeimprensa.ibge.gov.br/pt/noticias?view=noticia&id=1&busca=1&idnoticia=2382>> Acesso em 20 de maio de 2015.

Portal Brasil. **Cerca de 48% dos brasileiros usam internet regularmente.** 2014. Disponível em: <http://www.brasil.gov.br/governo/2014/12/cerca-de-48-dos-brasileiros-usam-internet-regularmente>. Acesso em: 12/01/2016

PRESSMAN, Roger S.. **Engenharia de software:** Uma abordagem profissional. 7. ed. Porto Alegre: Editora Apliada, 2011. Disponível em: <<http://pt.slideshare.net/lindomar2012/engenharia-de-software-7-edio-roger-spressman-captulo-1?related=1>> Acesso em: 13 de janeiro de 2016

RECO'RDON, David. **Tornado: Facebook's Real-Time Web Framework for Python.** 2009. Disponível em: <https://developers.facebook.com/blog/post/301>. Acesso em: 10 de janeiro de 2016

REIS, C. R. **Python na Prática:** Um curso objetivo de programação em Python, abril 2010. Disponível em: < <http://www.async.com.br/projects/pnp/pnp.pdf> >. Acesso em: 17 outubro 2015.

REZENDE, Denis Alcides; ABREU, Aline Franca de. **Tecnologia da informação:** Aplicada a sistemas de informação empresarial. 9. ed. Brasil: Atlas, 2013.

RODRIGUES, Andréa. **Desenvolvimento para internet.** Curitiba, PR: Editora LT, 2010. p. 9.

SANTANA, O. e GALESI, T. Python e Django. São Paulo: Novatec, 2010.

SILVA, Maurício Samy. **Criando sites com HTML:** sites de alta qualidade com HTML e CSS. Novatec, São Paulo, Brasil, 2008.

STAIR, Ralph M.; REYNOLDS, George M.. **Princípios de Sistemas de Informação:** Uma abordagem gerencial. 9. ed. São Paulo: Thomson, 2006.

SOMMERVILLE, Ian. **Engenharia de software.** São Paulo: Pearson Education do Brasil, 2003.

TANENBAUM, A. S. – **Redes de Computadores** – 4ª Ed., Editora Campus (Elsevier), 2003.

VENNERS, Bill. **The Making of Python.** 2003. Disponível em: <<http://www.artima.com/intv/pythonP.html>>. Acesso em: 07 de janeiro de 2016

XAVIER, Denys William. **O que é um algoritmo?.** 2010. Disponível em: <<http://www.tiexpert.net/programacao/algoritmo/o-que-e-um-algoritmo.php>>. Acesso em: 06 de janeiro de 2016

ZANETI JUNIOR, Luiz Antônio. **Sistemas de informação baseado na Tecnologia Web:** um estudo sobre seu desenvolvimento/ Luiz Antônio Zaneti Junior – São Paulo : FEA/USP, 2003.



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
- () Dissertação
- (X) Monografia
- () Artigo

Eu, **LEANDRO GONÇALVES DE ARAÚJO**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Sistema De Gerenciamento E Controle De Servidores Da Mikrotik** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 03 de Março de 2017.

Leandro Gonçalves de Araújo
Assinatura