Raspberry Pi: Uma solução para monitoramento e controle do tráfego de dados em micro e pequenas empresas.

Tiago Anony Ribeiro[1], Henrique Gabriel Gularte Pereira[1]

[1]Sistemas de Informação – Centro Universitário Franciscano Caixa Postal – 91.501-970 – Santa Maria – RS – Brazil

{tiago.a.ribeiro,ikkibr}@gmail.com

Abstract. *Companies even aware of the importance of technology to their presence in a globalized world, still fall short in one of the most important areas of information technology: security, for that, this project aims to bring a low-cost solution and great effectiveness for monitoring and data traffic control, thinking about current topics such as ecology and low power consumption for micro and small businesses.*

Resumo. *As empresas mesmo que cientes da importância da tecnologia para o sua presença em um mundo globalizado, ainda deixam a desejar em uma das áreas mais importantes das tecnologias da informação: a segurança, para isso, esse projeto visa trazer uma solução de baixo custo e grande eficácia, para monitoramento e controle de tráfego de dados, pensando em tópicos atuais como ecologia e baixo consumo de energia para micro e pequenas empresas.*

Palavra-chave: *Raspberry Pi, Firewall, Proxy, Raspbian, Tráfego de dados.*

1. Introdução

O uso da Tecnologia de Informação tem se tornado cada vez mais presente no dia-a-dia das organizações, o que leva a investimentos cada vez maiores dentro do setor empresarial. Esses investimentos variam consideravelmente de setor para setor da economia e de organização para organização [MARQUES, 2004], porém um dos setores com menor grau de investimento, principalmente em micro e pequenas empresas, é o da segurança da informação.

Isso ocorre devido ao alto valor necessário para se investir nessa área ou pela falta de recurso humano, para colocar o investimento em prática. Para tanto se faz necessária uma solução de baixo custo, que vise simplificar o processo de segurança da informação, e principalmente em monitoramento e controle de trafego de dados nas redes interna e externa, em micro e pequenas empresas.

Esse projeto propôs a criação de uma solução para esse problema que tem se demostrado cada vez mais presente nas empresas, a partir da instalação e configuração de um servidor de DHCP, proxy e firewall para controle e monitoramento de baixo custo, juntamente com o desenvolvimento de uma interface, visando simplificar as rotinas de administrador de redes, que será desenvolvida em Python. A instalação e configuração do servidor de DHCP, proxy e firewall, se dará através da utilização do projeto da Raspberry Foundation, denominado de Raspberry Pi. Um computador de baixo custo, compacto e com bom desempenho, que é capaz de prover os recursos necessários para suas tarefas de administração de redes, em micro e pequenas empresas.

O servidor irá utilizar sistema operacional baseado em kernel Linux, na sua distribuição Raspbian Whezzy, que foi desenvolvida exclusivamente para o projeto Raspberry Pi, baseada na distribuição Debian GNU/Linux, gerenciada pelo Projeto Debian foi oficialmente fundado em 16 de agosto de 1993 [DEBIAN, 2015].

Conforme pode ser visto na Figura 1, uma pesquisa realizada pelo Núcleo de Informação e Controle do .BR (NIC.BR) entre setembro e dezembro de 2013, sobre a proporção de empresas que tomaram medidas de ação sobre o uso da internet pelos empregados nos últimos 12 meses, que teve como base 6.159 empresas que declararam ter acesso a internet, 77% das empresas que possuem entre 10 e 49 pessoas empregadas orientaram os usuários sobre o uso da internet na empresa.

B12 - PROPORÇÃO DE EMPRESAS QUE TOMARAM MEDIDAS DE AÇÃO SOBRE O USO DA INTERNET PELAS PESSOAS OCUPADAS NOS ÚLTIMOS 12 MESES
Percentual sobre o total de empresas com acesso à Internet1

Percentual (%)		Orientou os usuários sobre o uso da Internet na empresa	Bloqueou o acesso a conteúdos de alguns ou todos os usuários	Monitorou os sites visitados por alguns ou todos os usuários	Monitorou o tráfego de dados individual de alguns ou todos usuários	Praticou outra forma de controle de alguns ou todos os usuários
Total		80	46	45	36	18
PORTE	De 10 a 49 pessoas ocupadas	77	40	38	30	15
	De 50 a 249 pessoas ocupadas	86	58	56	47	23
	De 250 ou mais pessoas ocupadas	94	83	80	73	33
REGIÃO	Norte	82	49	44	37	23
	Nordeste	78	45	44	31	18
	Sudeste	80	46	44	36	17
	Sul	78	41	46	38	15
	Centro-Oeste	81	52	51	40	24
MERCADOS DE ATUAÇÃO - CNAE 2.0	Indústria de transformação	77	41	43	36	17
	Construção	67	32	35	30	12
	Comércio; reparação de veículos automotores e motocicletas	84	51	48	37	20
	Transporte, armazenagem e correio	80	46	48	37	15
	Alojamento e alimentação	66	34	31	25	11
	Atividades imobiliárias; atividades profissionais, científicas e técnicas; atividades administrativas e serviços complentares	87	55	51	44	20
	Informação e comunicação	91	56	56	49	21
	Artes, cultura, esporte e recreação; outras atividades de serviços	82	45	44	37	17

Figura 1: Proporção de empresas que tomaram medidas de ação sobre o uso da internet pelas pessoas ocupadas nos últimos 12 meses. [CETIC, 2013]

Muitas vezes, a simples orientação não faz com que o usuário siga as regras previamente estabelecidas pela empresa. Por isso as próprias empresas devem monitorar e gerenciar sua rede.

Isso não quer dizer que as empresas devam simplesmente negar o acesso a informação através de bloqueios, que muitas vezes deixam de agregar valor para o colaborador ou a própria empresa, mas que deve ser realizado o estudo da situação e utilização dos recursos disponibilizados pela empresa aos seus colaboradores. Não que a

empresa deva somente monitorar, mas sim que: a empresa deve analisar todos os fatos, utilização, em que agrega e em que desagrega valor ao colaborador e para a mesma e, ver a melhor solução para cada fato, inclusive em alguns casos adotar bloqueio ao acesso de alguns recursos ou informações.

Outro objetivo do projeto, além de criar uma solução de custo acessível e grande utilidade para micro e pequenas empresas é proporcionar o monitoramento e controle da banda de internet e da rede através de ferramentas de monitoramento e gerencia de redes, bem como da geração de relatórios e de processos de configuração do ambiente de monitoramento. Tornando assim o acesso à internet mais dinâmico, rápido e de melhor distribuição para todos os colaboradores quando estiverem acessando, por meio dos recursos disponibilizados na empresa.

2. Referencial Teórico

2.1 Raspberry Pi

O Raspberry Pi é um computador no formato de cartão de crédito de baixo preço, que se conecta a um monitor de computador ou TV, e utiliza teclado e mouse como padrão [RASPBERRY, 2014].

Além de funcionar como um bom player de vídeo ligado à TV, a Raspberry Pi serve para montar servidores de baixo custo e com consumo levíssimo de eletricidade [COSTA, 2015].

Levando em consideração inúmeros aspectos, como dimensões, custo, economia, ecologia, energia, dentre outros que foi feita a escolha deste projeto para o desenvolvimento de um servidor de firewall para micro e pequenas empresas.

Para que se torne possível o desenvolvimento, um sistema operacional é instalado e configurado em um cartão SD (Secure Digital), que é inserido no slot já contido no projeto. A interface ethernet faz a comunicação do Raspberry Pi, para que o mesmo possa fazer o gerenciamento da rede. E para seu manuseio é utilizado um mouse e teclado sem fio ligados somente a uma porta USB e um monitor ligado a sua saída HDMI, sendo alimentado por sua saída de 5 Volts através de um cabo micro USB conforme Figura 2.

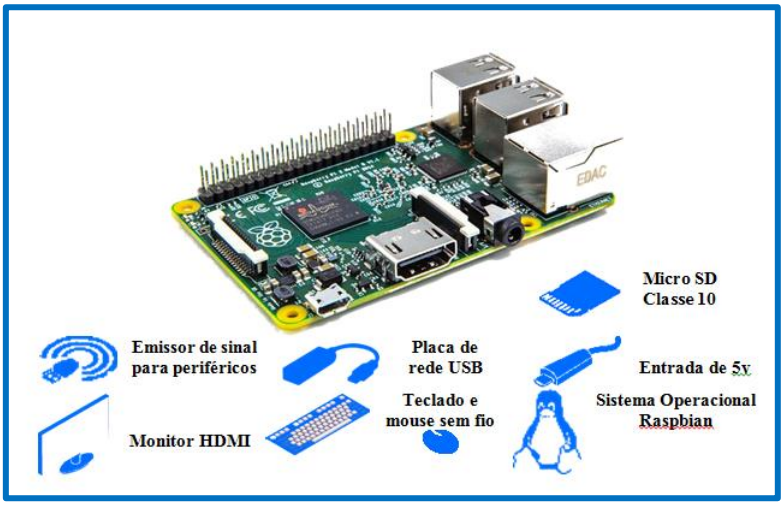

Figura 2: Elementos para funcionamento do Raspberry Pi 2 Modelo B

2.1.1 Especificações técnicas

Para este projeto foi utilizado o Raspberry Pi 2 modelo B, que conta com algumas melhorias quando comparado ao modelo "A" e aos modelos "B" e "B+", tais como o aumento em sua memória RAM, para 1(um) GB e clock de processador elevado para 900Mhz, e alteração de seu slot de cartão para micro SD, bem como o aumento de portas USB para 4(quatro), que já era disponibilizado em sua versão 1 no modelo "B+".

Quadro 1- Comparativo dos modelos lançados pela Raspberry Pi Foundation.

Especificação	Modelo A	Modelo B	Modelo B+	2 Modelo B
Preço	US$ 25	US$ 30	US$ 35	US$38
Processador	CPU: 700MHz; Low Power ARM1176JZ-F			CPU: 900Mhz, Arm Cotex-A7
Chipset	Broadcom BCM 2835			
Memória RAM	256 MB	512MB		1GB
Portas USB	1	2	4	
Saídas de vídeo	1 saída RCA e 1 saida HDMI.			
Dimensões	85.60 mm × 53.98 mm			
Armazenamento	SD / MMC / SDIO *slots* de cartão de memória			Mircro SD *slots* de cartão de memória
Rede	Sem interface	1 Interface ethernet 10/100MB		

2.2. Tecnologias utilizadas

O Raspberry Pi utiliza o Linux como sistema operacional. O Linux é tecnicamente apenas o *kernel*, e um sistema operacional é muito mais do que isso; a coleção total de *drivers*, serviços e aplicações compõem o sistema operacional. Uma diversidade de sabores (*flavors*) ou distribuições de Linux tem sido desenvolvida ao longo dos anos. Algumas das mais comuns em computadores *desktop* são Ubuntu, Debian, Fedora e Arch. Cada uma tem suas comunidades próprias de usuários e são ajustadas para aplicações específicas [RICHARDSON e WALLACE, 2013].

Em virtude do Raspberry Pi ser baseado em um *chipset* de dispositivos móveis, ele tem requisitos diferentes de um computador *desktop* [RICHARDSON e WALLACE, 2013]. Para este protótipo foram desenvolvidas algumas distribuições tais como: Pidora, Raspbian, Linux educacional Raspberry Pi da Adafruit, Arch Linux, Xbian, QtonPi, dentre outras. Essas distribuições em sua maioria foram baseadas em distribuições de sistemas operacionais para *desktop* e servidores que utilizam o *kernel* do Linux em sua base.

Para este projeto foi utilizado como sistema operacional o Raspbian que é, segundo Matt Richardson e Shawn Wallace em publicação na editora Novatec: a distribuição "oficialmente recomendada" da Fundação Raspberry Pi, juntamente com

alguns softwares e aplicações que ficam responsáveis pelo funcionamento dos serviços e servidores.

Em todos os serviços e servidores, foi pensado em utilizar as aplicações de maior credibilidade, estabilidade, robustez, sem mencionar que todas as tecnologias descritas a seguir, atualmente são as mais utilizadas na sua área de competência.

Para o gerenciamento de páginas da *web* e relatórios é necessária a utilização de um servidor HTTP. O Servidor Apache HTTP foi desenvolvido com a ideia de criar e manter um servidor HTTP de código aberto para sistemas operacionais modernos, incluindo UNIX e Windows NT. Visa principalmente fornecer um servidor seguro, eficiente e extensível que forneça serviços em sincronia com os padrões HTTP atuais [Apache, 2015].

Além do acesso a páginas da internet, é possível também a interpretação de arquivos HTML, PHP, ASP, dentre outros, proporcionando a visualização desses arquivos e o acesso a determinados serviços e aplicações, através de uma interface intuitiva.

Para garantir segurança nas transações HTTP, o servidor dispõe de um módulo chamado *mod_ssl*, o qual adiciona a capacidade do servidor atender requisições utilizando o protocolo HTTPS. Este protocolo utiliza uma camada SSL para criptografar todos os dados transferidos entre o cliente e o servidor, provendo maior grau de segurança, confidencialidade e confiabilidade dos dados. A camada SSL é compatível com certificados X.509, que são os certificados digitais fornecidos e assinados por grandes entidades certificadoras no mundo.

O protocolo *DHCP*, do inglês *Dynamic Host Configuration Protocol* (que ficaria, em português, algo como Protocolo de Configuração Dinâmica de Endereços de Rede), é um protocolo utilizado em redes de computadores que permite às máquinas obterem um endereço IP automaticamente. É um protocolo de serviço TCP/IP que oferece configuração dinâmica de terminais, com concessão de endereços IP de host, Máscara de sub-rede, Default Gateway (Gateway Padrão), Número IP de um ou mais servidores *DNS*, Número IP de um ou mais servidores WINS e Sufixos de pesquisa do *DNS*.

Resumindo, ele é responsável pela a distribuição automática de IP’s, para as estações clientes dentro da intranet, esse protocolo já está embutido na grande maioria dos modens, roteadores e *access point*.

Através de um servidor *DHCP* também foi possível designar um número específico para uma máquina qualquer na rede, sem que fosse necessária nenhum tipo de configuração na estação de trabalho, fazendo então, com que se possa aplicar de melhor forma normas de controle e monitoramento, como por exemplo, quando se utiliza um servidor de *proxy* e *firewall*.

O Servidor de domínio ou *Domain Name Server*(*DNS*), é responsável pela entrega e gerenciamento dos nomes dos dispositos na rede, para este trabalho foi utilizado o Bind, que é segundo a *Internet System Consortium Inc.* (*ISC*), o servidor de domínio mais utilizado no mundo.

Os protocolos *DNS* fazem parte dos padrões do núcleo da internet. Eles especificam o processo pelo qual, um computador pode encontrar outro computador,

com base em seu nome. A aplicação contém todo necessário para fazer e responder perguntas de serviço de nome. Contemplando além do *Domain Name Resolver* (resolvedor de nomes de domínio), uma série de ferramentas de diagnóstico e operacionais, alguns deles, tais como a popular ferramenta DIG, não são específicos para o Bind podendo ser usado com qualquer servidor de *DNS*.

Para melhor visualização dos acessos, bloqueios, tentativas de acesso, quantidade acessada, dentre outras opções. Foi necessário um gerador de relatórios do Squid. O *SARG* (Squid *Analysis Report Generator*) é uma ferramenta desenvolvida no Brasil, que permite à você ver para quais endereços seus usuários estão buscando internet através da análise do arquivo de log "access.log" do *proxy* Squid.

O *SARG* proporciona relatórios completos, mostrando ao seu administrador da rede, detalhadamente, separando por usuários informações tais como: quais sites foram acessados, em que horas, quantos bytes foram baixados, quantas conexões foram feitas. Resumindo todas as informações que um administrador de rede deve ter acesso, e ainda otimiza seu relatórios reportando de maneira clara: sites mais acessados, usuários que mais acessam, sites negados, falhas na autenticação dos usuários.

Todos esses relatórios são disponibilizados através de uma interface web que roda diretamente, nesse caso, do Raspbery Pi no seu servidor HTTP, onde é utilizada a solução Apache.

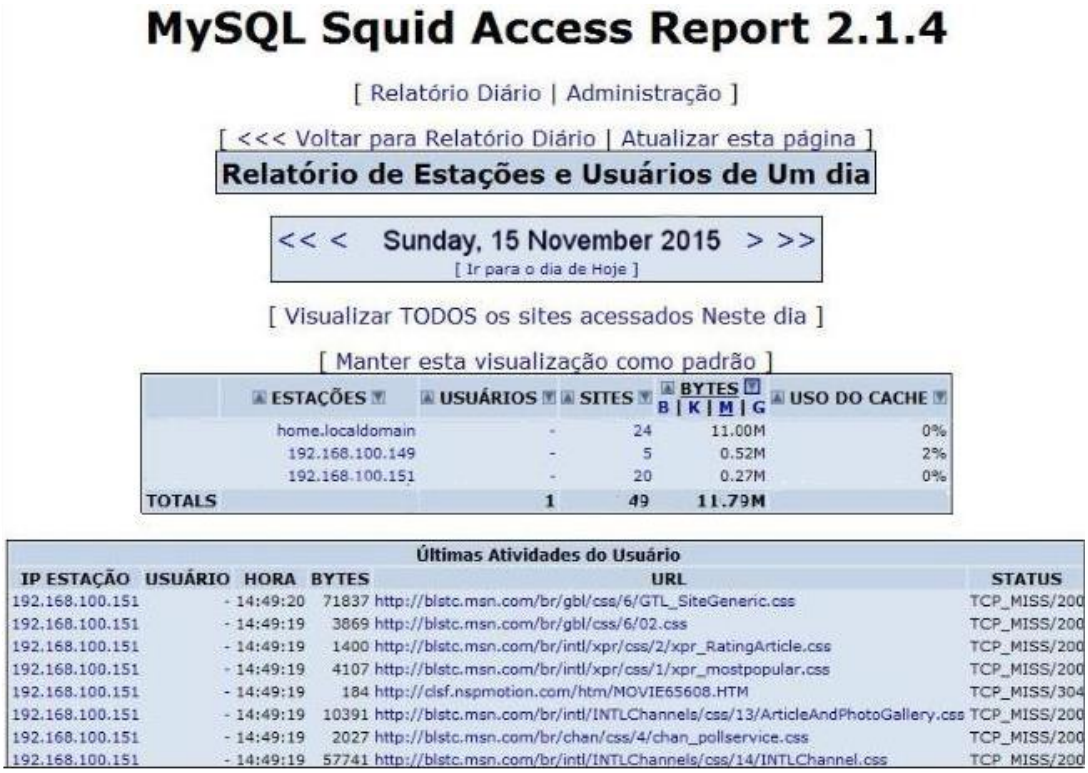

Figura 3. Exemplo de relatório do *SARG* (*Squid Analysis Report Generator*).

2.3. Segurança

Em um nível estratégico, a totalidade dos recursos de dados de uma empresa é quase insubstituível. Mesmo assim, dados nesse nível macro são amplamente negligenciados

pela diretoria corporativa. Em 2005, o Escritório de Administração e Orçamento dos Estados Unidos designou a segurança da TI como a "sexta linha dos negócios" [TURBAN ET AL., 2010]. É possível entender que a segurança é um assunto imprescindível para qualquer empresa. Nesse projeto foram utilizadas ferramentas de *firewall* e *proxy*, através do Raspberry Pi, para realizar o monitoramento, gerenciamento e controle das informações e dados que trafegam pela intranet da empresa.

Quando a empresa tem o monitoramento, controle e gerenciamento maior da intranet, esta torna se um ambiente mais eficaz e eficiente para a empresa, tornando o tempo do colaborador dentro da empresa melhor aproveitado.

2.3.1. Firewall

Ao falar sobre a informação digital em publicação ao site Techtudo em 2010, Fernando DAquino diz:

> *"Este ativo intangível consegue proporcionar diferenciais competitivos de grande magnitude o que implica diretamente a lucratividade das organizações."*

Para tanto, se faz necessária à utilização de ferramentas de monitoramento e controle de dados e informações, principalmente em ambientes corporativos. Uma das soluções com maior eficácia para proteção é o *firewall.*

Assim como a metáfora por trás do nome sugere, *firewall* é uma barreira de proteção que ajuda a bloquear o acesso de conteúdo malicioso, mas sem impedir que os dados que precisam transitar continuem fluindo. Em inglês, "*firewall*" é o nome daquelas portas antichamas usadas nas passagens para as escadarias em prédios. [MACHADO, 2012].

Podendo se apresentar nas formas de *hardware* e ou *softwares*, ferramentas de *firewall* ficam entre o link de comunicação e todos os dispositivos que acessam a rede, garantindo assim o monitoramento e controle das informações e dados que trafegam pela intranet, e que buscam acesso a internet. Para gerenciar o *firewall* foi utilizada a interface nativa do Linux, nas distribuições atuais, do iptables.

A interface iptables é a mais sofisticada já oferecida pelo Linux, tornando-o um sistema extremamente flexível para qualquer tipo de filtragem de rede que você possa fazer. Uma série de regras de filtragem pode ser agrupada de maneira que as tornam fáceis para que sejam testadas, ativadas e desativadas [PUDDY, 2006].

```
23  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
24  iptables -A INPUT -s 192.168.100.151 -j DROP
25  iptables -A INPUT -s 192.168.100.199 -j ACCEPT
26  iptables -A FORWARD -i eth1 -s 192.168.100.154 -m string --algo bm --string "facebook.com" -j ACCEPT
27  iptables -A FORWARD -i eth1 -m string --algo bm --string "facebook.com" -j DROP
```

Figura 4. Exemplo de como adicionar regras no iptables.

Na Figura 4, podemos analisar alguns dos comandos que são executados para configurar o iptables. Primeiramente o comando descrito na linha 23, faz com que a internet que chega através de um cabo UTP 5e, em sua interface ethernet eth0, possa ser compartilhada, com o restante da rede.

Nas linhas 24 e 24, temos respectivamente regras de bloqueio e aceite de

conexões oriundas de um determinado ip, sendo o IP descrito na regra que finaliza com DROP, o IP bloqueio e o que finaliza com ACCEPT.

Ainda acima nas linhas 26 e 27, é descrito uma liberação de acesso a domínio, no caso a rede social Facebook, ao IP 192.168.100.154 e, para o restante da rede todo e qualquer acesso a este domínio deverá ser bloqueado, impossibilitando o acesso de todas estações de trabalho, smartphones, notebooks, ou qualquer outro dispositivo que esteja usufruindo da rede corporativa.

2.3.1. Proxy

Proxy é o termo utilizado para definir os intermediários entre o usuário e seu servidor. E por isso desempenha a função de conexão do computador (local) à rede externa (internet). Como os endereços locais do computador não são válidos para acessos externos, cabe ao *proxy* enviar a solicitação do endereço local para o servidor, traduzindo e repassando-a para o seu computador.

Todas as requisições feitas ao servidor (o site que você quer acessar) passarão pelo seu *proxy*. Ao chegar ao site, o IP (Internet Protocol / Protocolo de Internet) do *proxy* fica registrado no cache do seu destino e não o seu. É pelo IP que os hackers conseguem invadir computadores, portanto deve-se manter o nível de segurança do seu *gateway* (porta de ligação com o *proxy*) seguro. Os riscos são vários, no entanto, dois deles podem ser enumerados como os mais fortes: ter seu computador invadido ou ter alguém navegando com o seu IP [BARWINSKI, 2012].

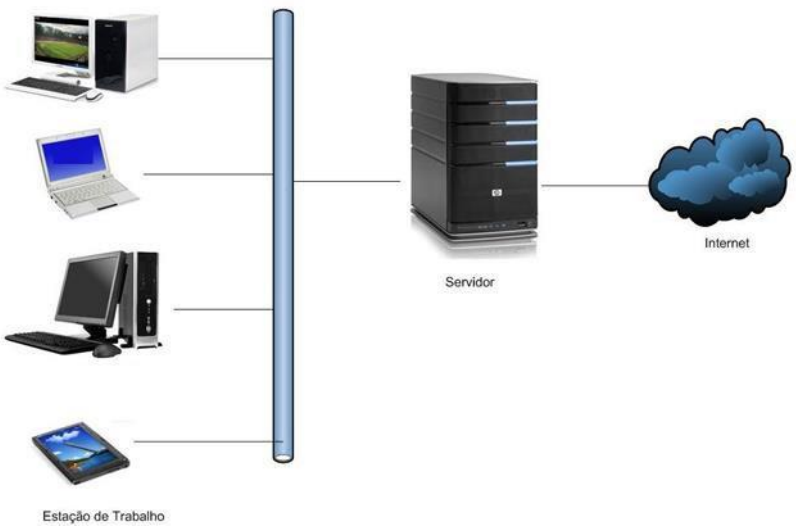

Figura 5. Funcionamento de um servidor *proxy*.

O Squid é um dos proxies para Linux, mais utilizados hoje na internet. Trata-se de um programa robusto, simples e extremamente confiável que se tornou um dos pacotes obrigatórios a serem instalados em qualquer provedor ou empresa que deseja aumentar a performance de sua conexão ou criar regras de acesso (ACL's) para servidores web [MARCELO, 2006].

O Squid adota todas as funcionalidades de um *proxy* bem desenvolvido e configurado, sendo ele um servidor HTTP com características especiais de filtragem de

pacotes, que normalmente são executados em um servidor de *firewall*. Ele fica aguardando por uma requisição de dentro do *firewall*, e repassa para outro servidor remoto do outro lado do *firewall*, depois simplesmente recebe a resposta do outro lado e a devolve para a estação cliente. Além disso, o Squid detém uma *cache*, configurável que pode armazenar sites, para aperfeiçoar o processo, apesar de *cache* ser visto sempre como um fator temporário, também é possível, por exemplo, deixar a página da empresa alocada na cache, para termos o acesso quase instantâneo nas estações clientes, e mesmo que ocorra a perda do sinal da internet, a mesma ainda pode ser acessada.

Há também a possibilidade de criação de regras, capazes de determinar permissões de acesso para: sites, IP's, certas palavras, dentre outras. Vale frisar que normalmente esse controle é feito pelo servidor de *firewall*, e quando usado em conjunto com esse tipo servidor, é de extrema valia a criação de uma regra que determina quais IP's, por exemplo, que não tem permissão para acessar nada na rede ou até mesmo, quais IP's que não passaram pelo servidor de *firewall*, deixando assim sua navegação e acesso a livres, porém abrindo uma rachadura na segurança proporcionada pelo mesmo.

2.4. Python

Para facilitar a administração das regras de controle para o tráfego de dados na rede interna e externa, foi pensado no desenvolvimento de uma interface simples e intuitiva, que propicie até mesmo a usuários leigos administrar sua rede.

Tendo em vista a abordagem, constante nesse trabalho, de software livre e linguagens atuais que foi feita a escolha pelo Python, uma linguagem de programação poderosa. Ela tem estruturas de dados de alto nível eficientes e uma abordagem simples, mas eficaz para programação orientada a objetos. Python possui sua sintaxe elegante e tipagem dinâmica, juntamente com a sua natureza interpretada, tornam Python ideal para scripting e desenvolvimento rápido de aplicações em muitas áreas [Python, 2015].

Por possuir fácil integração com linhas de comando, via terminal, Python proporciona todos os recursos disponíveis no *kernel*, de maneira rápida, de fácil compreensão e com baixo custo implementação. Tendo sua delimitação de blocos de instruções, não há o uso de delimitadores "{ }" utilizados nas linguagens *C* e *Java*, ou até mesmo o uso de ";" para determinar o fim da instrução, tornando assim o código mais limpo, de fácil compreensão e organizado.

Segundo Lutz, Python foi projetada para otimizar a produtividade do desenvolvedor, a qualidade do software, a portabilidade do programa e a integração dos componentes. Por mais que Python tenha sido criada com o intuito de ensinar programação de computadores, não é uma linguagem limitada ou com poucos recursos, atualmente dispõe de inúmeras bibliotecas, que asseguram a ela, ser uma linguagem de altíssimo nível, ou seja, bem mais próxima do raciocínio humano do que da arquitetura da máquina, completa e de grande usabilidade.

Juntamente com as bibliotecas nativas de Python, para melhor desenvolvimento do código, são importadas as bibliotecas *sys*, que auxilia no controle de execução do programa, *subprocess*, que possibilita de maneira simples e rápida a execução de comandos no terminal e *click*, que é responsável, nesse projeto, pela interface humano computador. A codificação *cp1252*, descrita na Linha 1, possibilita o uso de acentuação do dicionário português brasileiro conforme demonstra Figura 6.

```
1  # -*- coding: cp1252 -*-
2  import click
3  import sys
4  import subprocess
```

Figura 6. Codificação e importação de bibliotecas do arquivo gerenciadorDeRedes.py.

3. Metodologia

Estudos mostraram que o software, como todos os sistemas complexos, evoluem durante um período de tempo e os requisitos do negócio e do produto mudam frequentemente a medida que o desenvolvimento prossegue dificultando um caminho direto para um produto final [PRESSMAN, 2006].

Sendo assim, para o desenvolvimento deste projeto foram levados em consideração modelos de Engenharia de Software, tendo como principal, o modelo prototipação. Este modelo propõe uma abordagem dinâmica com grande interação com o cliente, realizando a construção de protótipos ao longo do projeto, os quais são entregues aos clientes, para poder ser realizados testes no ambiente.

A prototipação oferece um grande número de vantagens importantes, podendo ressaltar que: todo o requisito de sistema não tem que ser completamente determinado antecipadamente e pode ser trocado durante o curso projeto, a entrega da prototipação clara, definições de sistema entendível e especificações para o usuário final.

Estes fatores fazem, segundo a IBM, em 2002, isso possível para rapidamente testar o ambiente de desenvolvimento voltado para a funcionalidade, performance, interface com bancos de dados, etc.

Segundo Pressman, o desenvolvimento de um projeto seguindo os moldes da prototipação se dá através de muitas interações com o cliente, fazendo com que, antes do refinamento do projeto, todo o protótipo seja analisado, garantindo assim a satisfação do cliente. O projeto só é considerado finalizado e pronto para que seja realizada sua engenharia após muitas etapas de refinamento do protótipo, conforme Figura 7.

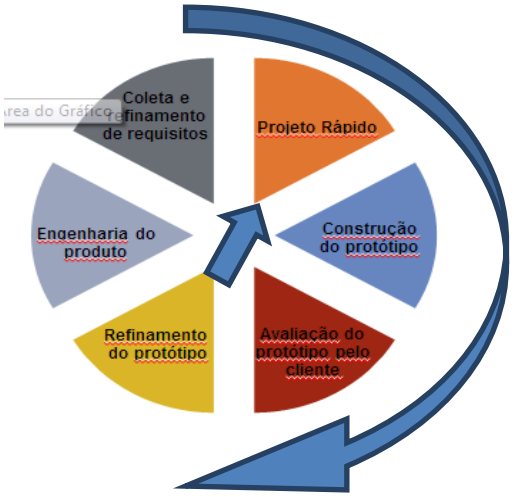

Figura 7. Modelo de processo de prototipação.

4. Desenvolvimento

Para ser possível o desenvolvimento da solução de monitoramento e controle de tráfego de dados nas redes de micro e pequenas empresas, que tivesse baixo valor de investimento, baixo consumo e acima de tudo, que propicie a segurança desejada pelos administradores da empresa, optou-se por tecnologias livres e estáveis.

Como solução de hardware, foi utilizado o Raspberry Pi em conjunto com uma placa de rede ethernet, que é conectada ao dispositivo através de USB. Mesmo o Raspberry Pi, já possuindo uma placa de rede, para maior controle e segurança do firewall, a solução demanda de duas placas redes, o que foi disponibilizado através do disposto USB/ethernet, Figura 8. Sua placa de rede onboard será conectada através de um cabo UTP, categoria 5e, ao modem que proverá a internet, e sua outra placa de rede, conectado então atráves de uma de suas quatro portas USB, conectará um cabo UTP, de mesma categoria, ao distribuidor de sinal, podendo ser ele: access point, switch, entre outros.

Figura 8. Placa de rede *ethernet* com conexão USB

Para softwares além de utilizar as aplicações de gestão de firewall iptables, gestão de proxy Squid e gestão de relatórios SARG, e o compilador de linguagem Python, bem como suas bibliotecas com o a adição de uma biblioteca não nativa, o click, que proporciona uma interface interativa com o usuário, foi desenvolvido então um gerenciador de redes que busca ser um modo de simplificar as regras aplicadas, principalmente ao iptables, visando assim tornar a solução de grande usabilidade, e a tornando acessível para os mais diversos públicos, sem requerer um conhecimento mais aprofundado sobre nenhuma dessas ferramentas.

```
31  def add_block_url():
32    """ADICIONAR BLOQUEIO DE URL"""
33    end = raw_input("Digite o endereco que deseja bloquear: ")
34    comando = "iptables -A FORWARD -i eth1 -m string --algo bm --string % s -j DROP" % end
35    print subprocess.check_output(comando, shell=True)
36
37    with open("/root/.firewall/firewall.sh", "r+") as arq:
38        conteudo = arq.read()
39        conteudo.replace("#marcador#", "#marcador#\n"+comando+"\n")
40        arq.seek(0)
41        arq.write(conteudo)
42
43    print "Comando executado."
44    click.pause("Pressione ENTER para continuar")
```

Figura 9. Função de bloqueio de URL do gerenciadorDeRedes.py

Na Figura 9, temos a parte do código referente a função que é responsável por adicionar bloqueio de URL's (páginas da internet). Na linha 33 é realizada a interação com usuário administrador, onde é impresso na tela "Digite o endereço que deseja bloquear", e a url será acrescentada no comando do iptables descrita na linha seguinte. A linha 35 utiliza a biblioteca subprocess, que fica responsável pela execução desse comando no terminal. Nas linhas 37 a 41 são realizados os procedimentos, para salvar esses comandos em um shell script, para isso primeiramente o arquivo é aberto em modo de leitura e escrita, então seu conteúdo é armazenado em uma variável (linha 38), após ele busca pela por #marcador#, e substitui no arquivo firewall.sh, por *#marcador# seguido de \n (para ir para próxima linha do arquivo) e o comando com mais um \n*.

Para remover um bloqueio, é realizado o comando que foi utilizado para adicioná-lo, com o acréscimo da argumentação "--delete" ao final do mesmo, para assegurar a remoção definitiva do bloqueio, o arquivo firewall.sh é aberto em modo de leitura e escrita e o comando é removido, como demonstrado na linha 85, da Figura 10.

```
76  def remove_block_url():
77    """REMOVER BLOQUEIO DE URL"""
78    end = raw_input("Digite o endereco que deseja remover o bloqueio: ")
79    comando = "iptables -A FORWARD -i eth1 -m string --algo bm --string %s -j DROP --delete" % end
80    comandoatualiza = "iptables -A FORWARD -i eth1 -m string --algo bm --string %s -j DROP" % end
81    print subprocess.check_output(comando, shell=True)
82
83    with open("/root/.firewall/firewall.sh", "r+") as arq:
84        conteudo = arq.read()
85        conteudo.remove(comandoatualiza)
86        arq.seek(0)
87        arq.write(conteudo)
88
89    atualizascript = "./root/.firewall/firewall.sh"
90    print subprocess.checkoutput(comando, shell=True)
91    print subprocess.checkoutput(atualizascript, shell=True)
92    print "Comandos executados."
93    click.pause("Pressione ENTER para continuar")
```

Figura 10. Função de remoção bloqueio de URL do gerenciadorDeRedes.py

Na Figura 11, é possível visualizar das telas do Gerenciador de Redes, desenvolvido em Python, onde na primeira imagem é adicionada a regra de bloqueio do IP 1.1.1.1, e na imagem a seguir, quando o usuário solicita a listagem das regras, através da opção 7 (sete), percebe-se a palavra DROP na mesma linha do IP 1.1.1.1, determinando que todo acesso oriundo desse IP, deverá ser bloqueado.

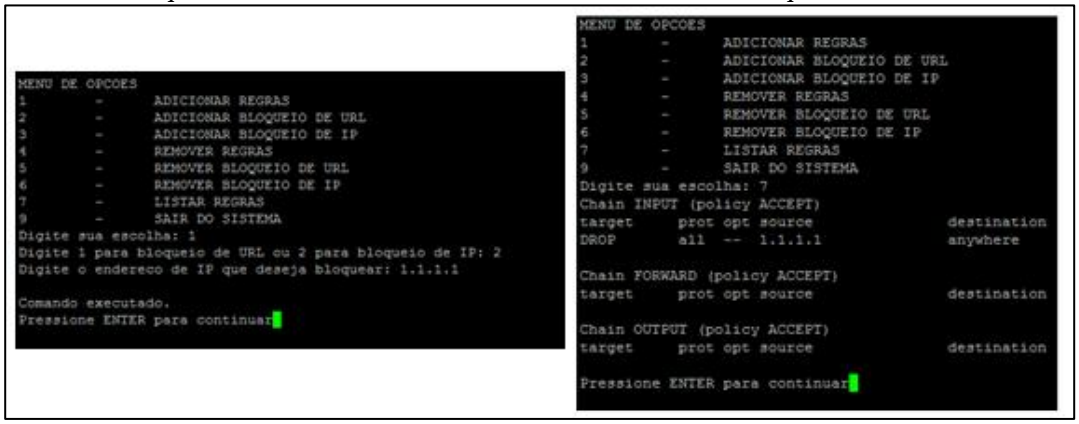

Figura 11. Telas de do Gerenciador de Rede.

O Raspberry Pi, é acessado por meio de um teclado e um monitor *HDMI* ou até mesmo por uma conexão *SSH*, tornando assim sua conexão, mais segura e estável, tornando o espaço físico utilizado pelo Raspberry Pi reduzido, e eliminando a utilização de periféricos no mesmo. O usuário administrador tem no menu de sua aplicação, as opções que são descritas na Figura 12, no diagrama de caso de uso, da aplicação desenvolvida em Python.

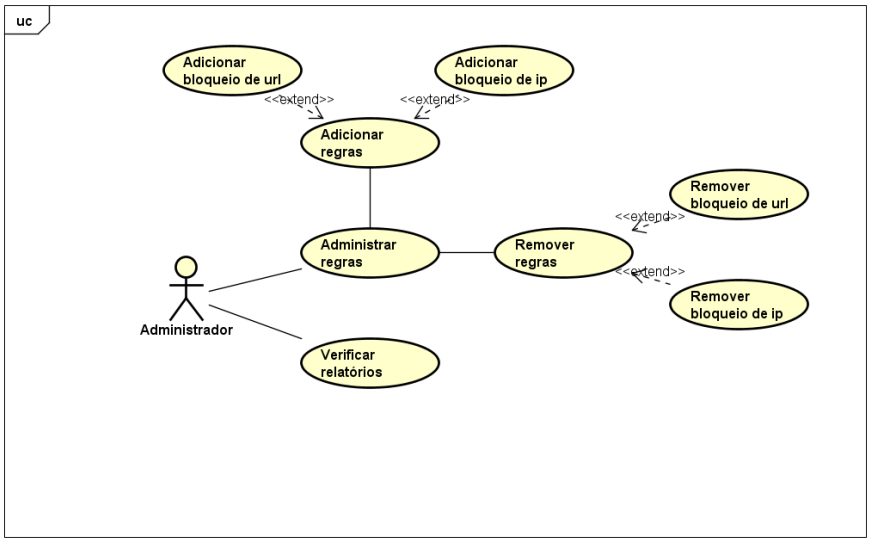

Figura 12. Diagrama de Casos de Uso

Para gerenciar a entrega de IP's aos clientes é um utilizado no Raspberry Pi, as funções de um servidor *DNS*, que é proporcionado através da aplicação Bind, em conjunto com um servidor de *DHCP*, responsável pela entrega automática dos números de identificação do protocolo IPv4, sem que seja necessária configuração na estação cliente, para acesso à rede.

Todo, e qualquer acesso a internet ocorrerá, somente mediante a gerência de monitoramento e controle de acesso implementada no Raspberry Pi, fazendo assim a intranet um ambiente mais seguro, profissional e devidamente monitorado. Garantindo a atomicidade das informações vitais para a empresa.

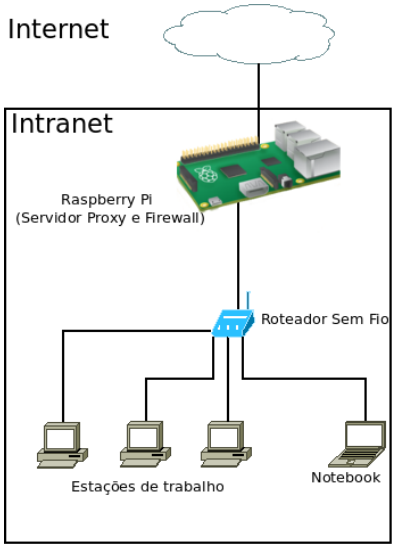

Figura 13. Utilização do Raspberry Pi como servidor *proxy* e *firewall*.

Um tutorial para instalação e configuração do Raspberry Pi, o código fonte desenvolvido e os arquivos de configuração estão disponíveis em https://github.com/tiagoanony/TFG/.

Vale frisar que as informações de instalação e configuração do Raspberry Pi como servidor de DNS, DHCP, Proxy e Firewall, tem sua garantia de funcionalidade apenas no cenário descrito nesse artigo.

6. Conclusão

Este projeto visa se tornar um recurso de grande importância para a segurança dos dados de micro e pequenas, sendo necessário um baixo investimento e com baixo consumo de energia, proporcionando assim uma solução economicamente viável, ecologicamente correta, eficaz e eficiente.

O Raspberry Pi, mesmo tendo suas configurações limitadas, atende ao esperado, quando usado em conjunto com o sistema operacional estável, devidamente desenvolvido para ele, em conjunto com um configuração otimizada, ser utilizado com a funcionalidade que lhe é proposta.

A solução atende ao esperado, se tornando peça de grande valia, para micro e pequenas empresas, que buscam uma solução eficiente, eficaz e de baixo custo, sendo possível por meio dessa, o controle de acessos à websites, ou conexão à internet através de softwares, otimizando assim a rede intranet e o uso da internet.

Sugere-se ainda para trabalho futuro, um estudo de caso detalhado dos resultados, desta solução em micro e pequenas empresas, propondo também trabalhos que demonstrem o uso do Raspberry Pi como outros tipos de servidores ou a utilização da placa como ferramenta para estudo de eletrônica, automatizando processos ou mesmo desenvolvendo protótipos de autômatos.

7. Referências Bibliográficas

APACHE (2015). About Apache. http://httpd.apache.org/docs-project. Acesso em Agosto de 2015

BARWINSKI, Luísa (2008). "O que é *proxy*?" em Tecmundo. http://www.tecmundo.com.br/navegador/972-o-que-e-proxy-.htm. Acesso maio de 2015

COSTA, Eric (2015). "Crie um servidor com o raspberry pi" em INFO online, editora Abril. http://info.abril.com.br/dicas/redes/crie-um-servidor-com-o-raspberry-pi.shtml. Acesso maio de 2015.

DAQUINO, Fernando (2010). "Profissão: Especialista em Segurança da Informação" em TechTudo http://m.tecmundo.com.br/seguranca/5366-profissao-especialista-em-seguranca-da-informacao.htm>. Acesso maio de 2015.

DEBIAN (2015). A Brief History of Debian. Debian Documentation Team. https://www.debian.org/doc/manuals/project-history/ch-intro.html. Acesso maio de 2015.

IBM; (2002) Practicing Object-Oriented Analysis and Desing- ERC2.2.;IBM Education ans Training.

LUTZ, Mark (2006). Python Guia de Bolso. 3ª. Edição, editora Alta Books.

MACHADO, Jonhatan (2012). "O que é firewall?" em Tecmundo. http://www.tecmundo.com.br/firewall/182-o-que-e-firewall-.htm. Acesso maio 2015

MARCELO, Antonio (2006). Squid – Configurando o Proxy para Linux. 5ª Edição, editora Brasport Livros e Multimídia.

MARQUES, Érico Veras (2004). "O Uso da Tecnologia de Informação nas Organizações: Um Estudo no Varejo de Moda no Brasil." Fundação Getúlio Vargas – Escola de Administração de Empresas de São Paulo. http://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/2597/74550.pdf. Acesso abril de 2015.

NIC.BR (2013). "Proporção de empresas que tomaram medidas de ação sobre o uso da internet pelas pessoas ocupadas nos últimos 12 meses. Percentual sobre o total de empresas com acesso à Internet." http://www.cetic.br/tics/empresas/2013/geral/B12. Acesso em maio de 2015

PEREIRA, Ana Paula (2009). "O que é DHCP?" em Tecmundo. http://www.tecmundo.com.br/2079-o-que-e-dhcp-.htm. Acesso maio 2015

PRESSMAN, Roger S (2006). Engenharia de software. 6ª Edição. Rio de Janeiro, RJ: McGraw Hill.

PRESSMAN, Roger S (2011). Engenharia de software. 7ª. Edição. Rio de Janeiro, RJ: McGraw Hill.

PURDY, Gregory N. (2006). Linux Iptables: Guia de bolso. 2ª Edição, editora Alta Books

PYTHON (2015). About Python. https://www.python.org/about. Acesso em agosto de 2015.

RASPBERRY.org. "What is a Raspberry Pi." Raspberry Pi Foundation. https://www.raspberrypi.org/help/faqs. Acesso em abril de 2015.

RICHARDSON e WALLACE. Primeiros passos com o Raspberry Pi. Matt Richardson e Shawn Wallace. Editora Novatec. https://www.novatec.com.br/livros/raspberrypi/capitulo9788575223451.pdf>. Acesso em maio de 2015.

TURBAN ET AL. (2010). Tecnologia da Informação Para Gestão: Transformando os Negócios na Economia Digital. 6ª. Edição. Efraim Turban, Dorothy Leidner, Ephraim McLean e James Wetherbe, editora Bookman.