MARLON CESAR BATISTA RODRIGUES

IMPLEMENTAÇÃO DE UM SISTEMA RESIDENCIAL DE CONTROLE DE ACESSO À INTERNET UTILIZANDO O *RASPBERRY PI* 3

FACULDADES UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI-MG
2018

MARLON CESAR BATISTA RODRIGUES

-

IMPLEMENTAÇÃO DE UM SISTEMA RESIDENCIAL DE CONTROLE DE ACESSO À INTERNET UTILIZANDO O *RASPBERRY PI* 3

Monografia apresentada ao Curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.
Área de Concentração: Redes.
Orientador: Prof. Marinho Soares de Souza.

FACULDADES UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI-MG
2018

rede de ensino
DOCTUM

FACULDADES UNIFICADAS DE TEÓFILO OTONI
NÚCLEO DE TCC / SISTEMAS DE INFORMAÇÃO
Autorizado pela Portaria 4.012 de 06/123/2004 – MEC

FOLHA DE APROVAÇÃO

A monografia intitulada: *Implementação de um sistema residencial de controle de acesso à internet utilizando o Raspberry PI 3,*

elaborada pelo aluno Marlon César Batista Rodrigues,

foi aprovada por todos os membros da Banca Examinadora e aceita pelo curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial da obtenção do título de

BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Teófilo Otoni, 5 de dezembro de 2018

Professor Orientador: Marinho Soares de Souza

Professora Examinadora: Yvssa Carneiro Desmots Eliote

Professor Examinador: Luiz Fernando Alves Souza

LISTA DE ILUSTRAÇÕES

RESUMO

Este trabalho acadêmico ressalta a importância em reduzir ao máximo a exposição das crianças e adolescentes aos riscos do acesso irrestrito a conteúdo da *internet*, considerando pesquisas recentes quanto ao uso da *internet* cada vez mais precoce pelo público infanto-juvenil. Desta forma, foi implementada uma ferramenta de controle, utilizando o *raspberry pi* como um servidor de *internet*, possuindo um filtro para bloquear conteúdos inapropriados. Na implementação foi utilizado o sistema operacional *Raspbian* que dispõe de recursos para acesso à *internet*. Também foi realizado pesquisa quanto às funcionalidades e recursos da placa *raspberry*, sua compatibilidade com ferramentas a serem utilizadas, bem como as configurações de arquivos de conexão com o provedor e o compartilhamento da *internet* na rede local, sendo necessário para isso, configurar um roteador em modo *bridge*. Foi também realizada a aplicação de ferramenta para implementação de filtros de controle como o *Squid* e a implementação de script para alinhar os navegadores ao *proxy squid* do servidor *raspberry*. Por fim, foi utilizada uma interface gráfica amigável, proveniente de integração com monografia apresentada nas Faculdades Unificadas Doctum de Teófilo Otoni por Márlon Francisco dos Santos em 2017, sendo adaptada ao servidor raspberry para que o administrador possa interagir com o sistema. Como proposto, a implementação do sistema se mostrou relevante e eficiente, os testes práticos realizados com sucesso e, apresentando uma ferramenta de baixo custo, comparando-se com ferramentas concorrentes já existentes no mercado. Finalizadas as configurações e implementado o sistema, obteve-se uma ferramenta de controle de acesso à internet para uso residencial.

Palavras-chave: *Raspberry PI*; *Raspbian*; *Squid, proxy*; *internet*; *script,* interface gráfica.

SUMÁRIO

INTRODUÇÃO

O presente trabalho tem como objetivo realizar a implementação de um sistema residencial de controle de acesso à internet, utilizando o *raspberry pi* como servidor, visando reduzir os riscos no acesso a conteúdo indevido, voltado especialmente para a utilização da internet por crianças e adolescentes.

Com o advento da internet, surgiram as mídias sociais oferecendo benefícios como aproximar pessoas através de redes sociais e tornando cada vez mais simples o acesso a informação. O entretenimento interativo, a comunicação entre pessoas e o comércio eletrônico são alguns dos serviços mais utilizadas pelos usuários na internet. No entanto, a utilização incorreta desta ferramenta de comunicação associada ao desconhecimento e à falta de orientação podem acarretar riscos.

Existem na internet muitos criminosos à procura de vítimas em potencial. Sem qualquer dificuldade, crianças e adolescentes têm acesso à internet, expondo em bate-papos informações pessoais, e isso é extremamente grave. Até chegar à fase adulta, os adolescentes estão em constante desenvolvimento e vulneráveis a riscos como: *ciberbullying*, chantagens on-line, conteúdos agressivos e considerados impróprios para sua faixa etária, alvos de adultos mal-intencionados, pedofilia, etc (SOUZA; OLIVEIRA, 2016, p. 3).

Segundo Thiago Tavares, presidente da *Safernet*, uma organização não governamental, “os casos de vazamento de *nudes* (fotos de nudez) não param de crescer ano a ano”. Pesquisadores do comitê gestor da internet no Brasil, revelaram que adolescentes de 11 a 17 anos tiveram contato com assuntos sobre automutilação (13%) e 20% sobre como ficar mais magro.

Dentro deste contexto, foi levantada a seguinte pergunta problema: A implementação de um sistema utilizando o *raspberry pi* pode garantir maior segurança

às crianças e adolescentes no acesso à internet e, proporcionar aos pais um melhor controle dos conteúdos acessados por seus filhos?

Baseado no problema apresentado, foram levantadas hipóteses para a busca de uma solução adequada para a questão em pauta:

H0 – A implementação de um sistema residencial de controle de acesso à internet utilizando o *raspberry pi* 3 não seria viável, considerando-se o custo-benefício da aplicação;

H1 – A utilização do *raspberry pI* 3 para implementação de um sistema residencial de controle de acesso à *internet* seria viável, tendo em vista a possibilidade de se utilizar o sistema Linux *Raspbian* que possui ferramentas de acesso à *interne*t, configurações de rede *wireless*, bem como excelente capacidade para uso do tipo doméstico;

H2 – A implementação de um sistema residencial de controle de acesso à *internet* usando o *raspberry pi* 3 seria viável, visto que se pode configurá-lo como um servidor, utilizando serviços para prover segurança à rede;

H3 – A implementação de um sistema residencial de controle de acesso à *interne*t utilizando o *raspberry pi* seria viável, considerando-se que possivelmente necessitaria de outro *hardware* como um *switch* para possibilitar a implantação do sistema;

H4 – A utilização do *raspberry pi* 3 como eficaz alternativa para implementação de um sistema residencial de controle de acesso à internet seria viável, considerando-se a relação custo-benefício em comparação com os *desktops* e dispositivos similares ao *raspberry* como o *Rock* 64, *Hickey* 960 e a *Tinker Board*.

Possuindo como objetivo geral, implementar um sistema residencial de controle de acesso à internet para os pais utilizando o *raspberry pi,* objetivos específicos foram estabelecidos para nortear a implementação do sistema:

- Configurar o *Raspberry* instalando ferramentas para gravar a imagem do sistema operacional (*Raspbian*) a ser utilizado no dispositivo e aplicativos para facilitar o acesso via *SSH*, uma vez que será usado um ambiente para servidor, poupando recursos de processamento, memória e disco.
- Montar o ambiente de simulação (laboratório) e realizar as configurações de rede necessárias para iniciar a instalação e as configurações dos serviços como Apache, *firewall* e *proxy squid* no *Raspserver*.

- Realizar a implementação do *proxy squid* e de regras de *firewall* a fim de bloquear ou permitir conexões e proteger as máquinas da rede contra programas maliciosos e acessos não autorizados.
- Implementar uma *blacklist* para configurar o sistema residencial de controle de acesso à *internet* de maneira que possa identificar *IP's*, *e-mail* ou domínios que possam oferecer riscos no acesso à rede.
- Implantar o sistema de controle de acesso à *internet* utilizando o *raspberry pi* em um ambiente real.

Quanto aos meios, este trabalho é classificado como bibliográfico e laboratorial, tendo em vista que, através do estudo sistemático de bibliografias publicadas, foi possível compreender o assunto pesquisado, desta forma, aprendendo como utilizar os recursos disponíveis para a implementação do sistema proposto. Enquadra-se na área laboratorial, visto que os métodos estudados nas bibliografias específicas proporcionaram a execução de testes apropriados ao projeto. O presente trabalho também foi classificado como aplicado, por ter foco específico na área de tecnologia, implementando um sistema de controle de acesso para solucionar determinado problema

Do ponto de vista científico, a relevância deste trabalho consistiu em pesquisas utilizando-se de literaturas da área e através do conhecimento adquirido no curso de Sistemas de Informação. Importante ressaltar que este trabalha visa proporcionar aos pais um sistema residencial de controle de acesso à internet utilizando uma ferramenta de baixo custo.

O presente trabalho teve foco em redes como sua área de concentração, utilizando-se de um microcomputador e de ferramentas indispensáveis para a implementação do sistema residencial de controle de acesso à internet.

Após a finalização das configurações, foram realizados testes práticos, utilizando sites que constam na *blacklist* cadastrada e apresentadas análises referentes à funcionalidade e eficiência do sistema.

A monografia está estruturada em 3 capítulos: O primeiro apresenta a fundamentação teórica concernente ao objeto de estudo em questão; O segundo descreve os procedimentos de implementação do sistema; O terceiro descreve a realização de testes e análises decorrentes do mesmo.

1 REFERENCIAL TEÓRICO

1.1 *Raspberry PI* 3

Fundação *Raspberry* é uma instituição sem fins lucrativos fundada em 2009, com sede no Reino Unido, responsável por desenvolver um computador de placa única chamado *raspberry pi*. Trabalha para capacitar pessoas do mundo inteiro em produção digital, fornecendo conhecimentos suficientes para que possam ser capazes de interagir num mundo cada dia mais digitalizado. A fundação *Raspberry* fornece computadores de baixo custo e alto desempenho, divulgando seus trabalhos e educando pessoas, treinando educadores que possam orientar outras pessoas a aprender e criar coisas com computadores, desenvolvendo recursos gratuitos para ajudar as pessoas a aprender sobre computação, proporcionando a acessibilidade na computação e produção digital, preparando-os para o futuro profissional.[1]

Computador de placa única, o *raspberry pi 3 model B* possui um processador *quad-core* de 1.4GHz, baseado no Cortex-A53, suportando instruções de 64 *bits*. *Dual-band* de 2.4 GHz e *wireless* de 5 *GHz*. *Ethernet* mais veloz e capacidade *PoE* através de um *PoE HAT* separado. O *PoE HAT* é um pequeno acessório utilizado para alimentar a placa *raspberry* através do *Power over Ethernet*, necessitando estar conectado à rede por uma fonte *PoE* (equipamento para fornecimento de energia e internet), dispensando o tradicional cabo de energia.[2] O *raspberry pi 3 model B* utiliza um conjunto de instruções ARMv8, proporcionando um excelente desempenho final. Possui 1GB de memória RAM e GPU *VideoCore* IV. É um pequeno dispositivo que permite a exploração da computação para aprender a programar em diversas

[1] Disponível em: <https://www.raspberrypi.org/about/>.
[2] Disponível em: <https://www.raspberrypi.org/products/poe-hat/>.

linguagens. Capaz de navegar na *internet*, reproduzir vídeos com alta definição, realizar processamento de textos e jogar jogos, esse dispositivo tem a capacidade de interagir com o mundo exterior, podendo ser usado em diversos projetos de fabricantes digitais. O *raspberry pi 3* é utilizado para aprender sobre o funcionamento de computadores, manipulando e programando.[3]

É possível instalar o sistema operacional *Linux* e poder usar para funções básicas de escritório como documentos de textos e planilhas, além de poder usar para navegar a internet e realizar pesquisas.[4]

1.2 *Software* Livre

Também conhecido como código aberto (*open source*), é aquele *software* que fornece adequadamente a liberdade ao usuário para distribuir, copiar, executar, estudar, alterar ou melhorar o *software*, proporcionando maior flexibilidade para seu crescimento, resultando num código de melhor qualidade, sendo o acesso ao código-fonte um pré-requisito.[5] Para ser considerado um *software* livre é necessário respeitar as quatro liberdades essenciais aos usuários, que segundo o site oficial gnu.org, são elas: A liberdade de executar qualquer programa conforme desejar e destiná-lo para qualquer propósito; A liberdade de estudar o funcionamento do programa, adaptando às necessidades desejadas; A liberdade de redistribuir cópias com intuito de ajudar outros independente do propósito; A liberdade de distribuir a outros usuários cópias modificadas, proporcionando que toda comunidade se beneficie das mudanças.[6]

Existe uma variedade de *software* livre com grande distribuição para uso pessoal como o sistema operacional *Linux* (*GNU/Linux*), servidores de aplicativos *JBoss* da *Red Hat* e o Apache *Tomcat*, servidores da *web* como o Apache ou *Nginx*, banco de dados como o *MySQL*, *PostgreSQL* e o *NoSQL* e uma enorme variedade de linguagens de código aberto, *frameworks* e outras ferramentas.[7] "O *Linux* fornece uma plataforma que permite que os desenvolvedores de *software* alterem o sistema

[3] Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
[4] Ibidem.
[5] Disponível em: <https://www.gnu.org/philosophy/free-sw.pt-br.html>.
[6] Ibidem.
[7] Ibidem.

operacional como quiserem e obtenham uma ampla gama de formas de ajuda para criar os aplicativos que precisam" (BRESNAHAN; NEGUS, 2014, p. 16). Um *software* livre pode ser disponível para uso, desenvolvimento e distribuição comercial.

Segundo o site oficial *gnu*.org, para que um *software* seja considerado livre, "é necessário lançá-lo sob uma licença de *Software* livre. A Licença Pública Geral *GNU* (*GNU GPL*), a mais conhecida dentre as existentes e utilizada pela maioria dos programas *GNU*, apresenta a versão 3 como a mais recente". De acordo com Bresnahan e Negus (2014, p. 13), são características básicas da *GNU GPL* "os direitos do autor original, mantendo-os para seu *software*, a distribuição livre, podendo ser alterado e redistribuído desde que inclua o código-fonte e os direitos autorais mantidos".

GNU é um sistema operacional que respeita a liberdade de seus usuários para controlar sua computação, cujo objetivo principal e contínuo é oferecer um sistema compatível com o *Unix*, um *software* 100% livre.[8] Semelhante ao Unix, o *GNU* é um sistema operacional completo e integralmente composto de *software* livre, aplicativos, ferramentas de desenvolvimento, etc. Lançado em 1983 por *Richard M. Stallman*, conhecido como projeto *GNU* da FSF (*Free Software Foundation*), "criado para se tornar uma recodificação de todo sistema operacional *Unix* que poderia ser distribuído gratuitamente" (BRESNAHAN; NEGUS, 2014, p. 13).

O nome "*GNU*" é um acrônimo recursivo para "*GNU's Not Unix*! ", desenvolvido para trabalhar em conjunto de forma a fornecer um sistema *GNU* funcional, contém todos os pacotes oficiais de software *GNU*, incluindo software livre que não pertencente ao *GNU*.[9] Os pacotes *GNU* possuem todos os programas oferecidos pelo sistema operacional aos usuários, aplicativos orientados a usuários, utilitários, bibliotecas, ferramentas, inclusive jogos. O objetivo principal do *GNU* é fornecer *software* livre afim de executar todos os trabalhos que os usuários de computadores necessitarem realizar.[10]

[8] Disponível em: < https://www.gnu.org/philosophy/free-sw.pt-br.html>.
[9] Ibidem.
[10] Ibidem.

1.2.1 As quatro liberdades

Para ser considerado um *software* livre, um programa deve oferecer aos usuários as quatro liberdades essenciais, ou seja, a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o *software*. Essas quatro liberdades possibilitam aos usuários, seja individualmente ou coletivamente, o poder de controlar o programa e tudo que este oferece a eles.[11]

A liberdade de executar o programa nada mais é do que deixar qualquer tipo de usuário ou organização livres para usá-lo em qualquer tipo de sistema computacional, qualquer tipo de trabalho e finalidade, sem a obrigação de comunicar ao desenvolvedor ou entidade específica. Desta forma, o importante é o propósito do usuário, sem considerar a vontade do desenvolvedor. Portanto, quando um usuário que usufrui da liberdade para rodar o programa para seus propósitos o distribui a outros, estes também serão livres para executar o programa com seus propósitos, sem qualquer imposição para utilização.[12]

A utilização da liberdade de estudar o código-fonte e fazer alterações significa que o código-fonte deve estar acessível ao usuário, incluindo a sua versão modificada em lugar do original. Uma forma adequada de alterar um programa preservando a característica de *software* livre é agregar módulos e sub-rotinas livres.[13]

A liberdade de redistribuir conforme desejar significa que qualquer usuário ou organização é livre para redistribuir cópias, alteradas ou não, de forma gratuita ou cobrando uma taxa pela distribuição, seja a quem for e em qualquer lugar, não sendo necessário pedir autorização ou pagar pela permissão para fazê-lo.[14] Deve-se incluir na liberdade de redistribuir cópias, formas executáveis ou binárias do programa, bem como o código-fonte, tanto da versão modificada quanto da inalterada. Contudo, não sendo possível produzir uma forma binária ou executável, tendo em vista que algumas linguagens de programação não suportam este recurso, deve-se conceder a liberdade de se redistribuir nessas formas caso seja desenvolvido um meio de criá-las.[15].

---

[11] Disponível em: < https://www.gnu.org/>.
[12] Disponível em: < https://www.gnu.org/>.
[13] Ibidem.
[14] Ibidem.
[15] Ibidem.

1.2.2 *Linux*

Segundo afirma Bresnahan e Negus (2014, p. 6), “*Linux* é um sistema operacional de computador, um sistema operacional consiste no *software* que gerencia seu computador e permite que sejam executados aplicativos nele”. Conforme o mesmo autor, é um sistema de código aberto caracterizado por detectar e preparar *hardware* quando da inicialização do sistema *Linux*, carregando os *softwares* necessários para acessar os dispositivos de *hardware* específicos, deve controlar ao mesmo tempo vários processos em execução decidindo quando e quais têm acesso à *CPU*.

De acordo com o referido autor, um sistema *Linux* deve gerenciar memória, especificando como serão manipuladas. Fornece interfaces de usuários para acessar o sistema, controla sistemas de arquivos controlando a posse e acesso aos arquivos e diretórios, proporciona acesso e autenticação de usuário criando contas de usuário e definindo limites para cada uma. Um sistema *Linux* também oferece utilitários administrativos como adicionar usuários, gerenciar discos, instalar *softwares*, bem como monitorar redes, etc. Uma enorme variedade de ferramentas de programação estão disponíveis com o *Linux*. Atualmente os sistemas *Linux* estão ainda mais desenvolvidos, oferecendo recursos não apresentados pelos primeiros sistemas *Unix* (sistema base para o *Linux*). Tais recursos avançados são utilizados para o *shell*, trabalhar com discos, controlar serviços e configurar uma vasta variedade de servidores.

> Os modernos sistemas *Linux* agora vão além do que podiam os primeiros sistemas *Unix*, recursos avançados no *Linux*, frequentemente usados em grandes empresas incluem, *clustering*, fazendo vários sistemas aparecerem como um para o mundo exterior, virtualização para gerenciar recursos de computação de forma mais eficiente, computação em tempo real e armazenamento especializado (BRESNAHAN; NEGUS, 2014, p. 7).

1.2.3 *Raspbian*

O *Raspbian* é uma distribuição *Linux* desenvolvida para rodar no *raspberry pi*, considerada o sistema operacional padrão da *Raspberry Foundation*.[16] Além de ferramentas de acesso à *internet*, esse sistema possui intenso controle do *hardware* da placa, inúmeros *softwares* de desenvolvimento, uma distro (distribuição *Linux*) completa. É uma distro completamente funcional, possui um tipo de abordagem ideal para o usuário aprender desenvolvimento de *software*, configuração de redes *wireless*, *Linux* e tecnologia de forma geral.[17]

O *Raspbian* é capaz de atender as necessidades do usuário que precisa de um sistema operacional capacitado no *raspberry* para uso regular e do tipo doméstico. Tomando como exemplo, o *Raspbian* é mais ágil do que o Ubuntu *MATE*, uma vez que utiliza um cartão *microSD* de classe 10. Oferece inúmeras ferramentas de desenvolvimento como *Python* e Java para quem deseja programar com o *raspberry*. Em distros pesadas, o *Raspbian* demonstra desempenho excelente quando usado no *raspberry pi*.[18]

1.3 Redes

Segundo João Eriberto Mota Filho (2013, p. 38), redes de computadores são o conjunto de dois ou mais computadores que compartilham dados, recursos de *hardware* e troca de informações entre si. “Nos locais onde existem pelo menos dois computadores, é muito vantajoso que ambos sejam conectados formando uma pequena rede” (VASCONCELOS; VASCONCELOS, 2008, p. 2). Existem vários meios de interligação, cabos, linha telefônica, sinais de satélites, fibra óptica, ondas de rádio, etc. Qualquer envio ou recebimento de dados caracteriza troca de informação. Quando um computador utiliza o *HD*, impressora ou *drive* de disquetes de outra máquina, configura-se compartilhamento de recursos. Ainda, segundo o

[16] Disponível em: < https://www.raspbian.org/RaspbianAbout>.
[17] Ibidem.
[18] Disponível em:<http://painel.passofundo.ifsul.edu.br/uploads/arq/20160711175902455087226.pdf>.

mesmo autor, “um dos micros poderá armazenar arquivos no disco rígido do outro”.

De acordo com João Eriberto Mota Filho (2013, p. 42, 45), as redes possuem uma família de protocolos (*TCP/IP*, *UDP*, *ICMP*, etc), sendo estes, regras que determinarão como a rede funcionará. Existem vários tipos de redes como a *Local Area Network* (*LAN*), a *Metropolitan Area Network* (*MAN*) e a *Wide Area Network* (*WAN*), que são classificados conforme sua abrangência ou sua forma de interligação, como a *intranet*, *extranet* ou *internet*.

A LANs ou redes locais, são redes de curto alcance geográfico, privadas e normalmente limitadas a residências, organizações, condomínios com um ou mais blocos, etc, possuindo alguns quilômetros de extensão (MOTA FILHO, 2013, p. 45). Utilizada amplamente por estações de trabalhos em empresas, computadores pessoais para compartilhamento de recursos, como impressoras e troca de informações, como documentos, arquivos, etc. “As *LANs* têm características que as distinguem de outros tipos de redes: (1) tamanho, (2) tecnologia de transmissão e (3) topologia” (TANENBAUM, 2003, p. 18). Ainda como afirma Andrew S. Tanenbaum (2003, p. 19), as *LANs* possuem um tamanho restrito, com transmissão quase sempre via cabo e topologias como barramento ou anel, estáticas ou dinâmicas.

A *MAN* ou rede metropolitana, é uma rede de médio alcance e alta velocidade, chegando a 50 ou 100 quilômetros (MOTA FILHO, 2013, p. 45). Esse tipo de rede abrange uma cidade, tendo como exemplo, rede de televisão à cabo disponível nas cidades. De acordo com Andrew S. Tanenbaum (2003, p. 20), quando foi percebida a audiência em massa que a *Internet* atraiu, as operadoras de TV a cabo entenderam que com algumas mudanças no sistema, serviços da *internet* de mão dupla em partes não utilizadas do espectro poderiam ser oferecidas.

As redes geograficamente distribuídas ou *WANs*, são redes de grande alcance geográfico e sem limitações máximas de distância, possuindo baixa velocidade (MOTA FILHO, 2013, p. 45). Esse tipo de rede abrange frequentemente um país ou continente e possui um conjunto de *hosts* (máquina ou computador conectado a uma rede) com a finalidade de executar programas dos usuários, estando estas máquinas conectadas por uma sub-rede que é responsável pelo transporte de mensagens entre os *hosts* (TANENBAUM, 2003, p. 20 - 21).

1.3.1 Protocolo de Redes

Os protocolos são responsáveis por determinar como será o funcionamento de uma rede. Cada tipo de rede um conjunto de protocolos específicos que realizam funções individuais viabilizando o funcionamento de toda a rede. As famílias de protocolos mais conhecidas são *TCP/IP*, *NetBIOS, e IPX/SPX*. Apresentando-se como a família de protocolos mais antiga tem-se o *TCP/IP*, englobando os protocolos *IP*, *TCP*, *UDP*, *ICMP*, etc (MOTA FILHO, 2013, p. 42).

1.3.1.1 Redes TCP/IP

Segundo João Eriberto Mota Filho (2013, p. 50), “ a *Internet Protocol* (*IP*), sem dúvida, é o protocolo mais importante da família *TCP/IP*. É responsável por grande parte do sucesso no tráfego de dados nas redes”. Ainda, segundo o mesmo autor, atualmente, a versão 4 do *IP* (*IPv4*) é a mais utilizada. Entretanto, a versão 6 do IP (*IPv6*) já está sendo usado no mundo inteiro para substituir o *IPv4*.

Tanto a versão 4 quanto a 6 do protocolo *IP* são utilizados para realizar a entrega de pacotes entre máquinas e, suas principais características são: Protocolo não confiável, onde simplesmente é feita a tentativa de entrega dos pacotes sem realizar verificações que garantam a integridade dos dados após a entrega, a detecção e correção de erros, etc. O *IP* deve ser associado a outros protocolos para agregar recursos como confiabilidade ao transporte de dados (*TCP*), velocidade (*UDP*), entre outros (MOTA FILHO, 2013, p. 50).

1.3.2 *DHCP*

O Protocolo de Configuração de *Host* Dinâmico (*DHCP*) é um protocolo com a finalidade de atribuir endereços *IP* automaticamente em uma rede de computadores,

fornecendo informações para configuração de dispositivos como servidores, *desktops* ou dispositivos móveis, proporcionando a comunicação em uma rede utilizando o protocolo *IP*.[19] É um protocolo de serviço *TCP/IP* que oferece configuração dinâmica de terminais. Através do *DHCP* um servidor distribui diferentes endereços *IP* de forma automática em uma rede de computadores à medida que que são feitas solicitações de conexão com a rede (VASCONCELOS; VASCONCELOS, 2008, p.25). De forma simplificada, o protocolo *DHCP* atribui endereços *IP* utilizando um modelo cliente-servidor. Quando um cliente se conecta à rede ele envia um pacote com solicitação de configurações *DHCP*.

Segundo João Eriberto Mota Filho (2013, p. 83-84), o *DHCP* é um serviço que fornece os dados de uma rede para que um cliente possa participar da mesma. Alguns dados que poderão ser fornecidos, entre outros: Endereço *IP*, máscara de rede, *gateway*, endereços *IP* dos servidores *DNS* da rede, data e hora atualizados, etc.

1.3.3 *DNS*

Segundo João Eriberto Mota Filho (2013, p. 82-83), *Domain Name System (DNS*) é um serviço que realiza a conversão de nomes de máquinas para *IP* ou de *IP* para nomes. É um protocolo básico de uma rede *TCP/ IP* responsável por localizar e traduzir para números *IP* os endereços de sites digitados no navegador e vice-versa. O sistema de nomes de domínios é um protocolo utilizado por padrão pelos provedores de acesso para manter a conexão funcionando.

O *DNS* oferece um serviço de *cache* armazenando as últimas consultas nos servidores durante determinado período de tempo, desta forma, agilizando o acesso à rede, tendo em vista que não é necessário realizar uma nova consulta para um endereço previamente acessado. Um esquema de criptografia fornecido pela *DNSSEC* (*DNS Security Extension*) faz uso de chaves públicas e privadas para garantir a autenticidade dos endereços consultados, desta forma, evitando possíveis fraudes. Muitos servidores *DNS* realizam serviços de detecção de sites falsos ou infectados e também sistema de proteção parental, caso seja necessário

[19] Disponível em: <https://www.isc.org/downloads/dhcp/>.

bloquear sites de conteúdo adulto.[20]

Na verdade, toda máquina em uma rede *TCP/ IP*, em princípio, sabe o endereço *IP* de um ou mais servidores *DNS*. Então, quando é digitado o endereço de um *site* (*URL*) em um navegador, este, imediatamente, pergunta a um servidor *DNS* qual é o endereço *IP* correspondente àquele nome. Uma vez recebida a informação, o navegador se conecta com o endereço *IP* do servidor desejado (MOTA FILHO, 2013, p. 83).

1.3.4 *Firewall*

Como afirmam Bresnahan e Negus (2014, p. 704, 705), é um sistema que realiza a tarefa de controle de tráfego de entrada e saída em uma rede de computadores, permitindo a acessibilidade ou bloqueando determinados tráfegos de dados com base em um conjunto estabelecido de regras de segurança. Pessoas que se alternam em turnos para analisar o tráfego de uma rede fazem parte desse sistema de *firewall*. "*Firewall* não é uma máquina, e sim um conceito. Qualquer recurso empregado na segurança de uma rede, inclusive humano, fará parte do sistema de *firewall*" (MOTA FILHO, 2013, p.351). Todos os dados cujo destinatário seja um servidor ou áreas sensíveis em uma rede deverão passar pelo *firewall,* cuja tarefa se baseia em observação e/ ou ação, autorizando a passagem de pacotes de dados ou negando esse processo seguindo a política de segurança local.

Segundo João Eriberto (2013, p. 354, 356), o *Firewall* possui alguns elementos, como o *netfilter*, um filtro de pacotes capaz de manipular ou bloquear determinados tipos de tráfego. Também é constituído por *proxies* como o *squid*, *software* cuja principal função está no uso em servidores *proxy* capazes de suportar *HTTP*, *HTTPS*, *FTP* e diversos outros formatos. "É importante ressaltar que *proxies* não aceleram a *internet*, os aceleradores de *internet*, por ocasião da consulta a *sites*, são os *caches*. Há *proxies HTTP* que possuem *cache*, como o *squid*" (MOTA FILHO, 2013, p. 356).

Ainda, segundo o mesmo autor, existem vários tipos de *firewall,* dentre eles: o *Firewall* de proxy, que fornece serviços adicionais como armazenamento em *cache* e

[20] Disponível em: <http://canaltech.com.br/internet/o-que-e-dns/>.

segurança de conteúdo; *Firewall* com inspeção de estado, onde são permitidos ou bloqueados tráfegos conforme o estado, a porta e o protocolo, monitorando toda atividade desde o início da conexão até o seu fechamento, seguindo regras do administrador; *Firewall* de gerenciamento unificado de ameaças (*UTM*), podendo incluir gerenciamento em nuvem, concentrando-se em simplicidade e facilidade no uso; *Firewall* de próxima geração (*NGFW*), para bloquear ameaças modernas, tais como ataques na camada da aplicação e *malwares* avançados.

1.3.4.1 Proxy (squid)

O *squid* é um *proxy HTTP* 1.0 completo, repleto de recursos como controle de acesso, autorização e criação de log para desenvolvimento de aplicativos de *proxy* da *web* e de conteúdo.[21] Esse *proxy* possui diversas opções de otimização de tráfego, onde a maioria proporciona instalação simples e alto desempenho. “Existem *proxies HTTP*, *FTP*, *SMTP*, *POP3*, *DNS*, etc., para quase todos os serviços de aplicações”(MOTA FILHO, 2013 p. 356). Segundo o mesmo autor (2013, p. 357 - 358), um *proxy* de qualidade entende protocolos, sendo capaz de bloquear o tráfego de dados não adequado. Os *Proxies* atuam intermediando clientes e servidores de forma a evitar contato direto entre eles, desta forma aumentando o nível de segurança. O *Squid* é usado para realizar economia de tráfego na *Internet*, melhorar o desempenho, proporcionar uma navegação mais rápida aos clientes finais e fornecer conteúdo estático, dinâmico e de *streaming* para os usuários da *Internet* no mundo inteiro.[22]

> Um *proxy* deve saber se comportar como cliente e servidor porque é assim que ele funciona. Quando um *host* procura o *proxy* para solicitar um site, esse *proxy* receberá a conexão do *host* e, nessas condições, estará atuando como servidor. No entanto, quando o *proxy* sair em busca do *site*, ele o fará na qualidade de cliente (MOTA FILHO, 2013, p. 357).

[21] Disponível em: <https://www.squid-cache.org/intro/>.
[22] Disponível em: <https://www.apache.org/>.

1.3.5 Apache

O Apache é um servidor *web open-source* responsável por disponibilizar páginas e todos os recursos que podem ser acessados pelo usuário na *internet*. Como um servidor livre, ele pode ser alterado através do seu código-fonte por qualquer pessoa.[23]

Distribuído sob a licença *GNU*, o Apache executa funções como compras *online*, envio de mensagens e *e-mail*. O servidor Apache é compatível com o protocolo *HTTP* versão 1.13, dispondo do *mod_ssl*, um módulo que inclui ao servidor a capacidade de atender solicitações com protocolo *HTTPS*, utilizando a camada *SSL* para criptografia do tráfego de dados, desta forma, proporcionando maior segurança do tráfego entre cliente e servidor.[24]

Disponibilizado para *Windows*, *Novell Netware*, *OS/2* e outros sistemas do padrão *POSIX*, como o *Unix* e o *Linux*, possui funcionalidades que são mantidas por meio de uma estrutura de módulos, permitindo que os próprios usuários escrevam seus módulos através da *API* do *software*.[25]

1.4 *Shell Script*

Segundo Julio Cezar Neves (2017, p. 96), *Shell* é um interpretador de comandos e é uma camada entre o *kernel* e o usuário. Como linha de comandos do *Linux*, o *Shell* interpreta comandos digitados pelos usuários para chamar programas e automatizar tarefas. Toda sessão no *Linux* começa por uma solicitação de *login* e senha, podendo ser acessado como usuário comum ($) para atividades mais simples ou, como superusuário (#), também conhecido como *root*, possuindo um maior grau de privilégios.

Um *Shell script* possibilita combinar comandos para solucionar tarefas de alta complexidade. Tanto o usuário quanto o sistema, utilizando um arquivo de texto

[23] Disponível em: <https://www.apache.org/>.
[24] Idem.
[25] Ibidem.

executável, são capazes de executar uma sequência de operações, instruções e testes.

“Um programa *Shell*, também chamado de *script*, é uma ferramenta fácil de usar para construir aplicações “colando e aglutinando” chamadas de sistema, ferramentas, utilitários e binários compilados” (NEVES, 2017, p. 96).

De acordo com Bresnahan e Negus (2014, p. 147), *scripts* são utilizados durante a inicialização da maioria dos sistemas *Linux* para fazerem os serviços funcionarem através da execução dos comandos necessários, equivalem a arquivos em lote *MS-DOS*, podendo conter longas listas de comandos.

Scripts são grupos de comando utilizados na verificação e montagem de todos os sistemas de arquivo, carregando todos os serviços do sistema, manipulando os arquivos e diretórios. São também utilizados em configuração de *consoles*, redes e também fornecem tela de *login*, tendo como *Shell* padrão o *bash* (BRESNAHAN; NEGUS, 2014).

1.4.1 Principais Tipos de *Shell*

Conforme afirma Julio Cesar Neves (2017, p.100), “o *Shell* é um programa interpretador de instruções, que foi escrito em diferentes versões”. Dentre os diversos tipos de *shell* existentes, podem-se citar os principais como o *Bourne Shell* (*sh*) que é um interpretador de instruções padrão do *Unix*, sendo o mais utilizado.

O *Bourne-Again* (*bash*), também padrão do *Unix*, bastante conhecido pelo seu sistema operacional hospedeiro. Desenvolvido por *David Korn*, o *Korn Shell* (Ksh) reconhece todos os comandos do *Bourne Shell*, proporciona uma variedade de opções e instruções. Ainda como um dos principais, o C *Shell* (*csh*) é o mais usado nos ambientes *Berkeley* e *XENIX*, com estrutura de linguagem semelhante à linguagem C.

1.4.2 Executando o *Shell*

Segundo Bresnahan e Negus (2014, p. 64), três das mais comuns formas de utilizar a interface de *Shell* no *Linux* são o *prompt* de *Shell*, uma janela de terminal e o *console* virtual.

Para iniciar um *Shell*, a janela de terminal, também conhecida como emulador de terminal, pode ser aberta com a interface do *desktop Linux* rodando. Em distribuições *Linux* o terminal pode ser aberto clicando com o botão direito do *mouse* na área de trabalho, em seguida clicando na opção “Abra o Emulador de terminal aqui” ou, clicando no menu do painel, localizado na parte inferior ou superior da tela para executar o aplicativo (BRESNAHAN; NEGUS, 2014).

Ainda segundo Bresnahan e Negus (2014, p. 148), existem duas maneiras básicas de executar um *script* de *shell*:

- Utilizando o nome do arquivo como argumento para o *Shel*l (*bash myscript*);
- O nome do interpretador colocado na primeira linha precedido por “#!” (#! *bin/bash*).

Para a execução do arquivo com o *script* configurado utiliza-se a permissão através de comando específico (*chmod* +x nome_arquivo).

1.4.3 Principais Tarefas do *Shell*

As principais tarefas segundo Julio Cesar Neves (2017. p. 97, 98), em ordem de execução realizadas pelo *Shell* são:

- Exame de linha de comandos recebidas – Ocorre a identificação da linha de comando através da interpretação dos caracteres, considerando os espaços em branco;
- Atribuição – Identificada quando é encontrado um sinal de igualdade sem espaço em branco em ambos os lados do sinal;

- Resolução de redirecionamentos – Prepara o ambiente para que redirecionamentos identificados, seja de entrada (*stdin*), de saída (*stdout*) ou de erros (*stderr*), possam ser executados;
- Substituição de variáveis – Verificação da definição das variáveis existentes no escopo da linha de comando, substituindo por seus respectivos valores;
- Substituição de metacaracteres – Ocorre a substituição de metacaracteres (*, ? ou [ ]) por seus possíveis valores na linha de comando;
- Passa linha de comando para o *Kernel* – Após a execução das tarefas anteriores, a linha de comando é montada com todas as substituições realizadas e executada pelo *kernel*, gerando um número de processo (*PID ou Process identification*) e, depois de encerrado o processo é apresentado um *prompt* (interpretador de linha de comando), mostrando que está disponível para executar outro comando.

1.4.4 Comandos

O *Linux* oferece uma enorme quantidade de comandos para serem utilizados em linhas de comando. Após informar os comandos e pressionar a tecla <*ENTER*>, estes são examinados pelo *Shell* e executados pelo *Linux*, gerando a ação desejada ou a mensagem de erro (NEVES, 2017, p. 5).

Exemplo de comandos *Shell* segundo Júlio Cesar Neves:

$ *who* Só o comando

$ *who –H* Comando e opção

$ *who am i* Comando e argumento

1.4.4.1 Help - Ajuda

Comando utilizado para auxiliar na correta implementação das linhas de comando. O comando *help* é usado para mostrar informações gerais sobre os comandos não autônomos de *Shell*, sendo muito úteis para novos usuários (NEVES, 2017, p. 6).

Exemplo de ajuda de acordo com Júlio Cesar Neves:

$ *help*
Gnu bash, version 2.05b. 0 (*1*) *– release* **(***i386-redhat-linux-gnu*)
These Shell commands are defined internally. Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.

A star (*) nest to a name means that the command is disabled.

% [*DIGITS* | *WORD*] [*&*] ((*expression*))
. filename *:*
[*arg…*] [[*expression*]]
alias [*-p*] [*name* [*=value] . . .*] *bg* [*job_spec*]
. . . .

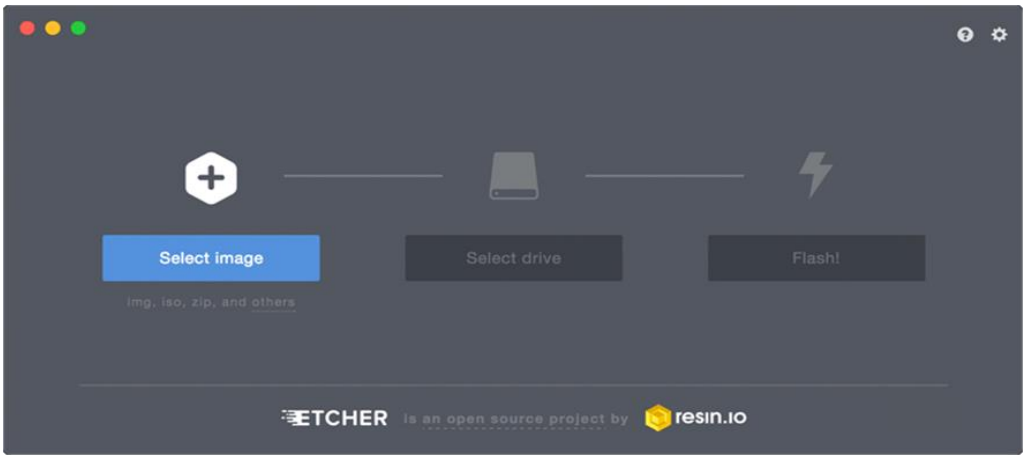
2 IMPLEMENTAÇÃO

Na execução deste trabalho foi utilizado uma máquina com a plataforma *Linux* Ubuntu. Foram utilizados na parte física *(hardware*) uma placa de microcomputador *raspberry pi 3 Model B* contendo quatro entradas *HDMI*, uma entrada de rede RJ45, uma entrada para cartão *microSD*, uma entrada *microUSB.* Ainda foram utilizados um roteador *D-Link*, um monitor *A*OC, um cabo *HDMI*, um mouse, um cabo adaptador *HDMI* para *VGA*, um adaptador *Usb* de rede RJ45 externa, um cartão microSD de 16GB, classe 10 e um adaptador de cartão *microSD*.

Para a implementação do sistema de controle de acesso à internet utilizando o *raspberry pi*, foi necessário acessar o site *raspberrypi.org*, na sessão de *downloads* e baixar o pacote .zip do *Raspbian Stretch Lite*, uma imagem do sistema operacional baseado no *Debian*, desenvolvido pela *raspberrypi.org* exclusivamente para ser utilizado no *raspberry pi*. Em seguida, no site do *etcher.io* foi baixado para a plataforma em utilização o *Etcher*, um utilitário gratuito desenvolvido pela *resin.io*, de código aberto e usado para gravar arquivos de imagem, como arquivos .iso, .img, .zip, dentre outras. Também foi utilizado o serviço *SSH* para acesso remoto do sistema a ser implementado.

Para iniciar a gravação da imagem do sistema operacional *Raspbian* no *raspberry pi,* foi necessário descompactar o arquivo executável do *etcher*. Utilizando-se de um cartão *microSD* de 16 Gb e um adaptador *USB* plugado no computador, foi gravada a imagem do *raspbian* executando a ferramenta *Etcher* conforme figura 1 a seguir.

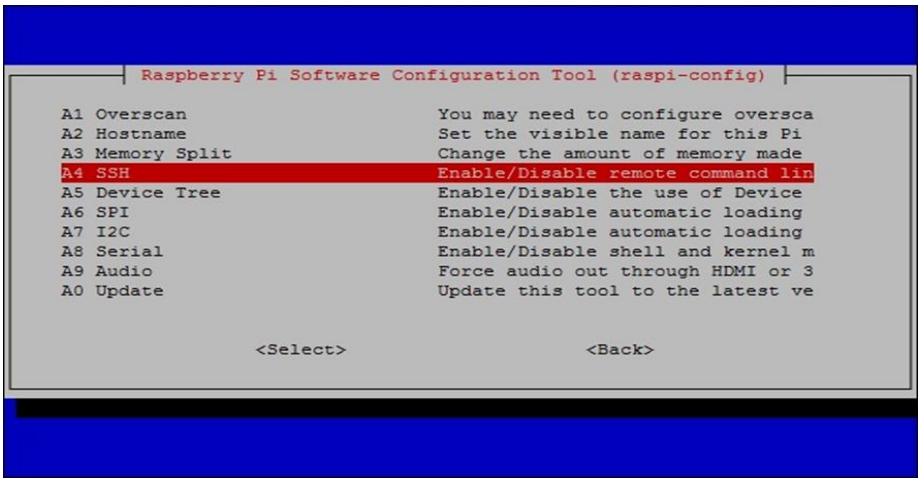
Figura 1 – *Etcher*, ferramenta de gravação da imagem *Raspbian*



Fonte: Do próprio autor.

Após o procedimento de gravação da imagem, o cartão foi inserido na entrada microSD da placa *raspberry* que foi ligada e conectada a um monitor via adaptador *HDMI* para *VGA*, iniciando o *raspbian* no modo terminal, utilizando o usuário “pi” e a senha “*raspberry”*, ambos como padrão. Em seguida foi digitado o comando “sudo *raspi-config* “ para habilitar o ssh do sistema e, conectado via cabo em um roteador da rede local, verificado o ip para realizar acesso remoto para as demais configurações. Na ferramenta de configuração do *raspberry pi* foi selecionada a opção 5, “*<interfacing options>”* e, clicando na opção *“SSH”*, em seguida, foi selecionada a opção “*< select >*” para habilitar o serviço de acesso remoto conforme figura 2.

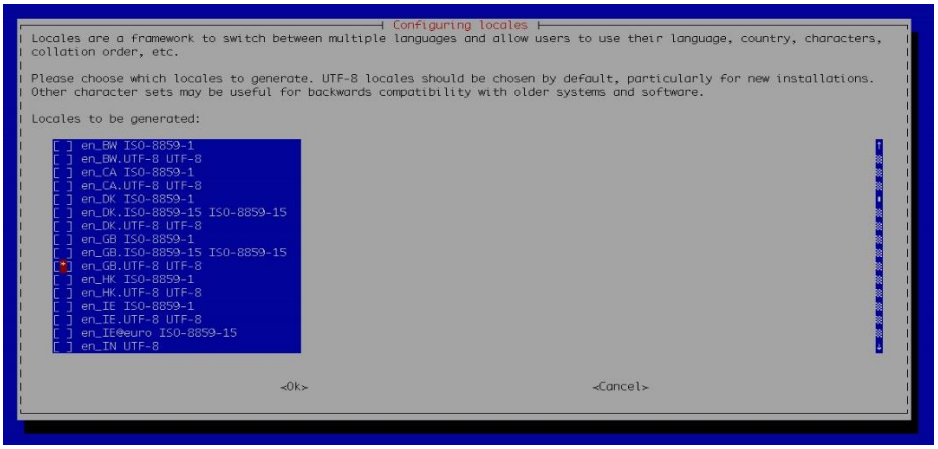
Figura 2 – Habilitação do serviço *SSH*



Fonte: Do próprio autor.

Prosseguindo com a configuração, foi acessado novamente a interface de configuração através do comando “sudo *raspi-config”* para selecionar o fuso horário, o idioma local e as ajustar o *layout* do teclado com o sistema do *raspberry pi*. Primeiramente, para definir o idioma foi selecionada a opção 4, “*<Localisation options>*” e, em seguida, clicando na opção “*<change locale>”,* conforme a figura 3, alterando a localidade “*eng_GB.UTF-8 UTF-8*” para as opções “*pt_BR ISO-8859-1* e *pt_BR.UTF-8 UTF*” com a barra de espaço e, por fim, na próxima tela, definindo o padrão do sistema com a opção “*pt_BR.UTF-8*” de acordo com a figura 3.

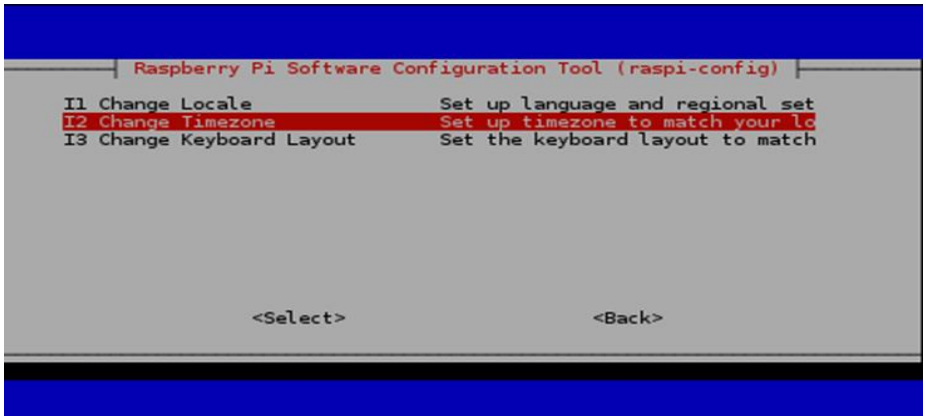
Figura 3 – Configuração do idioma local



Fonte: Do próprio autor.

Para selecionar o fuso horário, após selecionar “<L*ocalisation Options*>”, como na figura 4, foi selecionada a opção “*<Change Timezone*>”, selecionada a opção “America” e em seguida a cidade pertencente ao fuso horário local segundo figura 4.

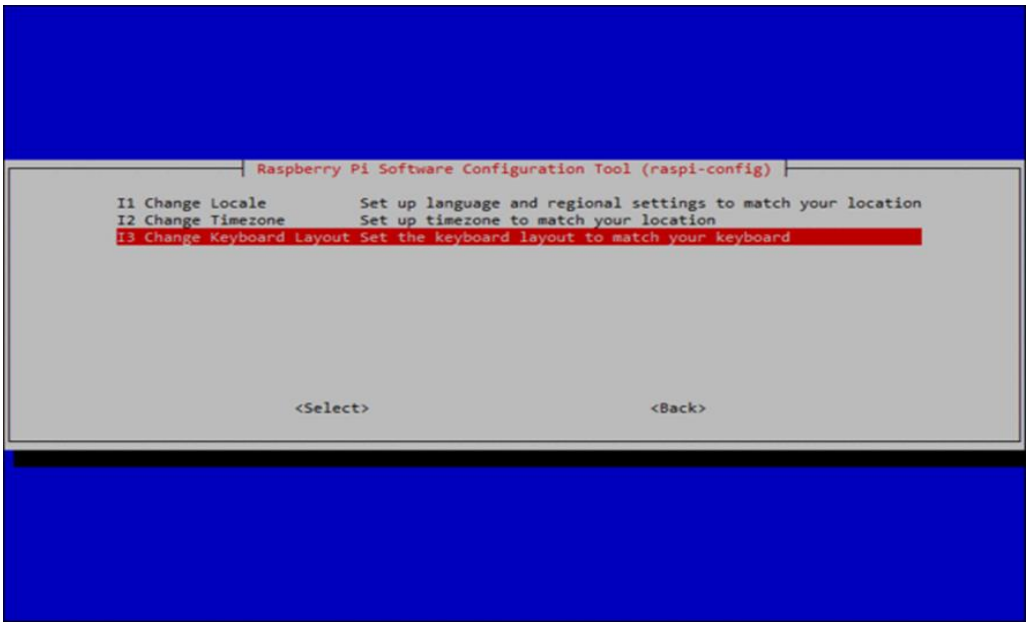
Figura 4 – Configuração do fuso horário local



Fonte: Do próprio autor.

Finalizando a parte de configuração do *raspberry pi*, após a opção "<*Localisation Options*>", foi selecionada a opção "<*Change Keyboard Layout*>" e definidas as configurações de acordo com o tipo de teclado utilizado conforme figura 5.

Figura 5 – Configuração do teclado utilizado



Fonte: Do próprio autor.

Prosseguindo com a implementação do sistema, a *raspberry pi* foi conectada ao roteador da rede local que serviu de laboratório para estabelecer a conexão com a *internet* e realizar testes. Inicialmente foi identificado o ip disponibilizado para a *raspberry* e este utilizado para fazer o acesso remoto com a placa, economizando recursos do sistema como a memória.

Para fazer o acesso remoto, foi digitada no *prompt* de comando (*shell*) da plataforma utilizada para os trabalhos a linha de comando abaixo e em seguida a respectiva senha de acesso do *raspberry*:

- *ssh pi@numero_ip*;
- *password*: **********

Próximo passo foi estabelecer a conexão da placa *raspberry* com o provedor de *internet* da rede local. Tendo em vista que a conexão estabelecida é do tipo *PPPoE*,

foi necessário baixar os pacotes *pppoeconf* e *ppp* para realizar as configurações e estabelecer a comunicação com a *internet*. *PPPoE* é a sigla em inglês para *Point-to-Point Protocol over Ethernet* que se refere a um protocolo de rede para conexão com tecnologia *Ethernet*, ligando uma placa de rede a vários usuários em uma rede *LAN* através de uma linha *DSL*.

No *shell* foram utilizados os seguintes comandos para instalação dos pacotes:

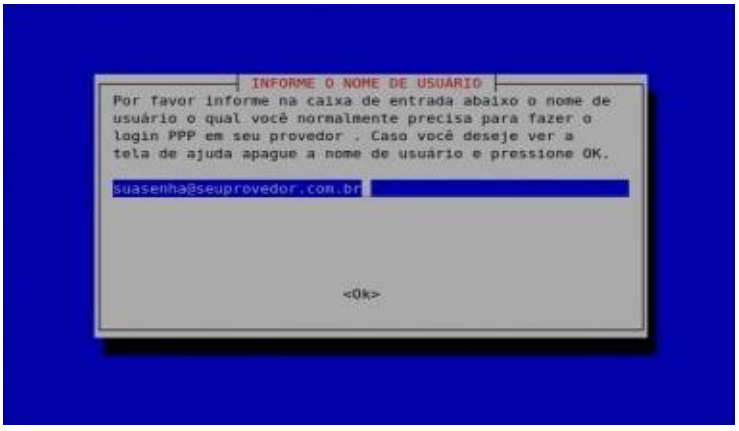
- *sudo apt update* – atualização do sistema da placa *raspberry pi*;
- *sudo apt install pppoeconf ppp.*

Realizados os procedimentos de instalação, deu-se início às configurações para estabelecer a conexão da placa com o provedor de *internet* digitando no *shell* em modo *root* o seguinte comando abaixo:

- *# pppoeconf*

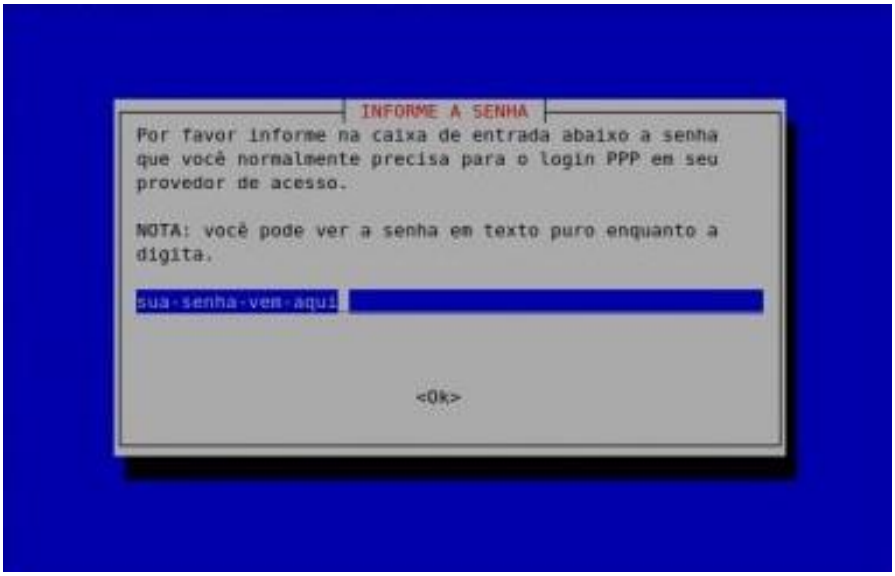
Após execução do comando "*pppoeconf*", surgiram várias telas de configuração da conexão *PPPoE*, dentre elas, a figura 6 que solicitou o nome do usuário e a figura 7 que solictou a senha de acesso ao provedor, bastando nas demais telas apenas confirmar clicando na opção "<Sim>" e na última tela a opção "<Ok>". Não foi necessário identificar a interface que comunica com o modem do provedor, durante a execução do pacote "*pppoeconf*" a interface eth0 foi identifcada automaticamente de acordo com a figura 6.

Figura 6 – Inserção do nome do usuario da conexão PPPoE



Fonte: Do próprio autor.

Figura 7 – Inserção da senha do usuário PPPoE

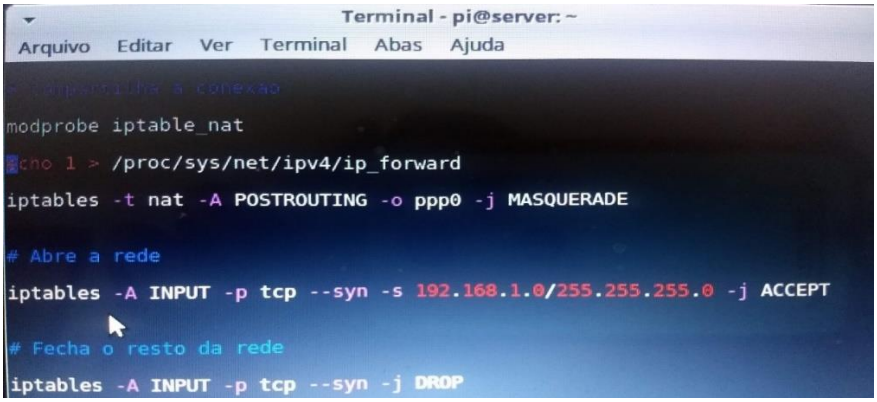


Fonte: Do próprio autor.

Dando prosseguimento à implementação do sistema de controle de acesso à internet, foram configurados os arquivos "*/etc/sysctl.conf*", "*/etc/network/interfaces*", "*/etc/rc.local*" e o arquivo "*/etc/resolv.conf*" para liberar o sinal da *internet* pela placa *eth1* do servidor para o restante da rede.

O arquivo "/etc/*rc.local*" é um script onde é definido o que deve ser executado logo após iniciar todos os serviços do sistema (popularmente conhecido como *boot*). Utilizando o *shell* no modo *root, n*este arquivo foram implementadas regras de mascaramento da interface "*ppp0*", regras de acesso e compartilhamento de *internet* para a rede local e regras de restrição de acesso conforme figura 8.

Figura 8 – Configuração do arquivo *rc.local*



Fonte: Do próprio autor.

Acessando o arquivo "*/etc/sysctl.conf*", através de um editor de texto, foi realizada uma configuração para permitir o roteamento de pacotes na rede, retirando o símbolo *hashtag* para descomentar a segunda linha de comando:

- *Editor /etc /sysctl.conf*
- *#net.ipv4.ip_forward=1*

Prosseguindo com a implementação do sistema, no arquivo "*/etc/resolv.conf*" foram definidos os servidores *DNS* que resolverão os endereços. Para acessar o arquivo foi utilizado um editor de texto inserido os *IP-DNS* do provedor de *internet.*

- *editor /etc /resolv.conf*
- *nameserver 8.8.8.8*
- *nameserver 128.201.101.72*

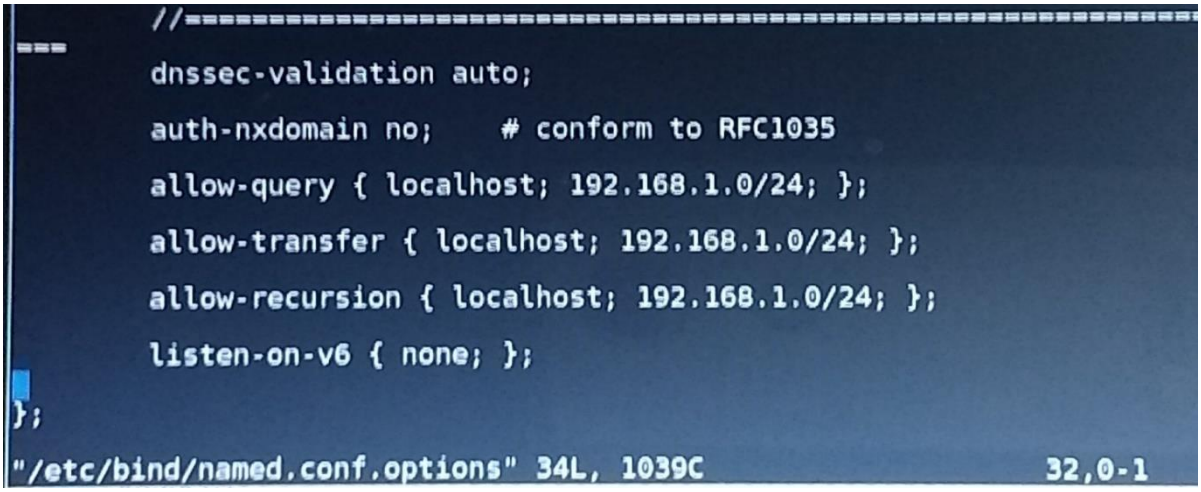
Para a configuração do arquivo "*/etc/network/interfaces*", antes foram necessárias as instalações de alguns pacotes, dentre eles, o *Bind9* e o *isc-dhcp-server.*

- *sudo apt update*
- *sudo apt install bind9 bind9utils*
- *sudo apt install isc-dhcp-server*

O *Bind* (*Berkeley Internet Name Domain*) é o servidor de protocolo *DNS* mais utilizado no mundo e extremamente importante na construção de servidores *Linux.* Através desta ferramenta é possível armazenar o endereço IP associado ao nome, indicar um servidor de nome autorizado para o um domínio, contém propriedades básicas do domínio e da zona do domínio, também contém o nome real do host a que o *IP* pertence.

Usando o editor de texto, de acordo com a figura 9, foram realizadas algumas configurações no Bind, dentre elas, alterações acessando o arquivo com o comando "sudo editor */etc/bind/named.conf.options*", desabilitando o protocolo ipv6, trocando o termo entre chaves de "any" para "none" e, inserindo linhas de comando para definir em quais interfaces da rede o *bind* irá funcionar.

Figura 9 – Configuração do arquivo *named.conf.options*



Fonte: Do próprio autor.

Com os respectivos comandos abaixo, após a execução das alterações, o *bind* foi reiniciado para possibilitar a resolução de nomes e permitir a verificação do *status* do servidor *DNS*:

- *sudo service bind restart*
- *sudo systemctl status bind -l*

Implementado o servidor *DNS*, foi também criado um domínio local (*raspserver.int*), servindo de suporte para as consultas do próprio domínio do servidor. Inicialmente foi realizada a alteração do arquivo "*/etc/hostname* da máquina e do arquivo "*/etc/hosts"*, trocando a nomeclatura *raspberrypi* pelo nome *server* no primeiro arquivo e acrescentando no segundo, além do nome *server,* o nome do domínio através das seguintes linhas de comando respectivamente:

- *sudo /etc /hostname*
- *sudo /etc /hosts*

De acordo com a figura 10, após as alterações, o arquivo "*/etc/hosts* ficou da seguinte forma:

Figura 10 – Configuração do arquivo *hosts*

Fonte: Do próprio autor.

Dando prosseguimento à implementação, utilizando um editor, no arquivo "*/etc/bind/named.conf*" foi comentado o arquivo "*/etc/bind/named.conf.default-*zones" conforme a figura 11, e criado um novo arquivo onde será configurada a zona local do domínio do servidor, utilizando a seguinte linha de comando:

- *include "/etc/bind/named.conf.internal-zones*";

Figura 11 – Configuração do arquivo *name.conf*

Fonte: Do próprio autor.

Em seguida, de acordo com a figura 12, foi aberto e configurado o arquivo da zona local utilizando-se de um editor de texto com o comando abaixo:

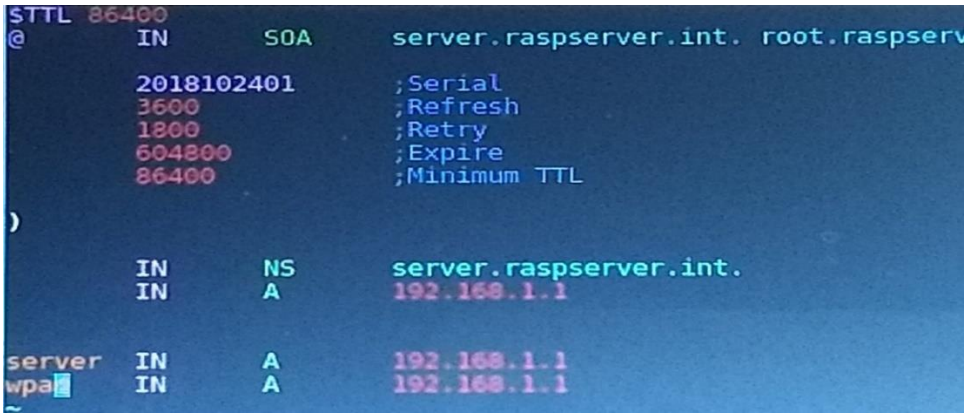
- *Sudo vim /etc /bind /named.conf.internal-zones*

Figura 12 – Configuração do arquivo *named.conf.internal-zones*

Fonte: Do próprio autor.

Para identificar os *hosts* das redes e os respectivos *IP's* através de uma consulta, foi criado um novo arquivo de configuração com o editor de texto com nome "*/etc/bind/raspserver.int.lan"* segundo figura 13:

Figura 13 – Configuração do arquivo *raspserver.int.lan*

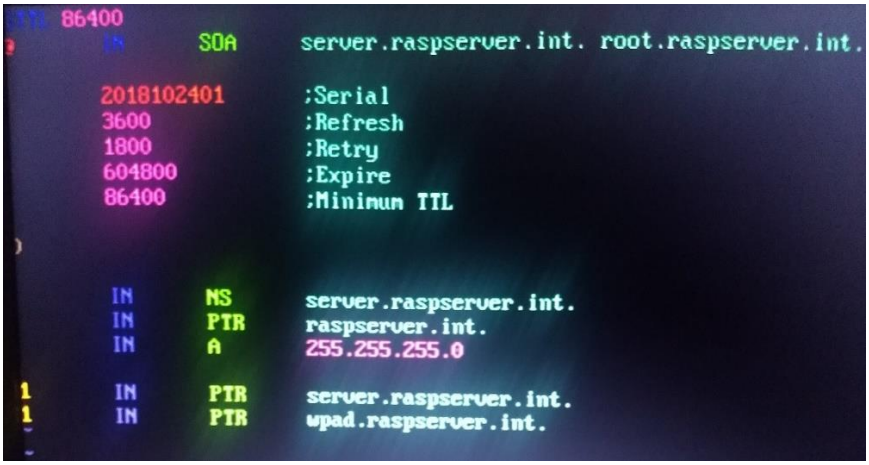


Fonte: Do próprio autor.

Além do arquivo de *hosts*, como mostra a figura 14, foi criado um arquivo de configuração reverso para identificar através dos *IP's* os nomes das máquinas utilizando a seguinte linha de comando:

- *sudo vim "/etc/bind/1.168.192.db*";

Figura 14 – Configuração do arquivo reverso



Fonte: Do próprio autor.

Ao final das configurações dos respectivos arquivos, a ferramenta bind foi reinicializada para começar a resolver os nomes na rede local.

- *sudo service bind9 restart*

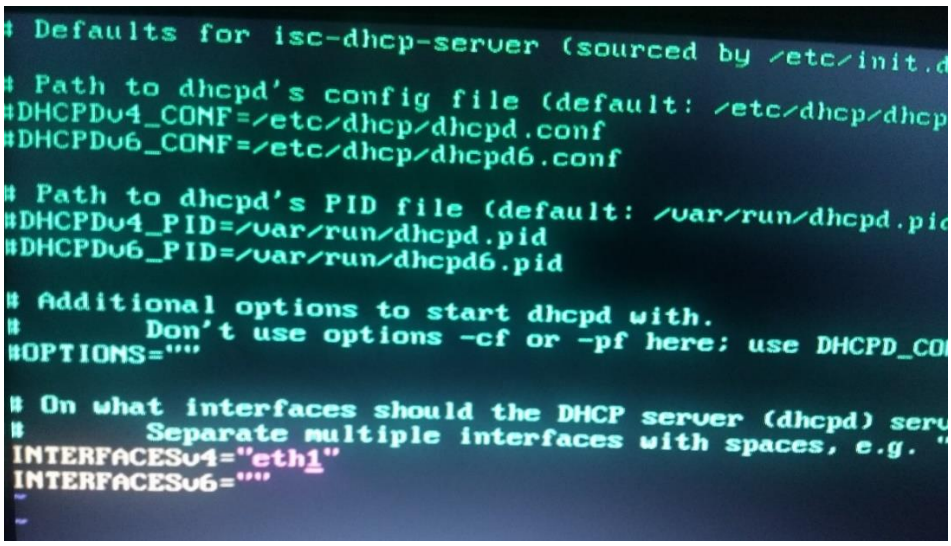
O *isc-dhcp-server* (*Dynamic Host Configuration Protocol*) é um protocolo desenvolvido e mantido pela *Internet System Consortium,* ele possibilita o envio via *broadcast* de informações de rede automaticamente, forncendo *IP's* dinamicamente para as máquinas da rede local. Para a proceder à instalação do serviço *DHCP,* foi utilizado no *shell* as seguintes linhas de comando:

- *sudo apt update*
- *sudo apt install isc-dhcp-server*

Para que o serviço *DHCP* funcionasse corretamente foi necessário editar o arquivo "*/etc/default/isc-dhcp-server*" para definir em qual interface de rede (*eth1*) o serviço *dhcp* irá responder, conforme figura 15 a seguir.

- *Sudo vim /etc /default /isc-dhcp-server*

Figura 15 – Definição da interface onde o *DHCP* será executado



Fonte: Do próprio autor.

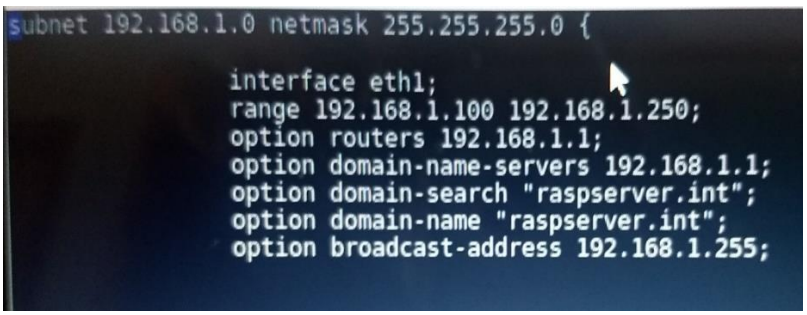
O próximo arquivo configurado foi o "*/etc/dhcp/dhcpd.conf*", onde será descomentada a linha #log-facility local17, possibilitando a realização de log's no servidor e em seguida criado um arquivo de configuração.

- *Include "/etc/dhcp/conf/rede.conf ";*

Em seguida, foi criado dentro do arquivo *etc um diretório chamado "/etc/conf "* e, dentro desse diretório foi criado um arquivo com o nome "rede.conf", onde será configurada a subrede *eth1* para distribuir faixa de *IP* para as máquinas da rede local, conforme figura 16.

- *sudo vim /etc /dhcp /conf*
- *cd /etc /dhcp /conf*
- *sudo touch rede.conf*
- *sudo vim rede.conf*

Figura 16 – Configuração da subrede (*eth1*)



Fonte: Do próprio autor.

Após realizar as devidas configurações no arquivo da subrede, o serviço *DHCP* foi reinicializado, pondendo ser verificado o status do serviço conforme comandos abaixo respectivamente.

- *sudo service isc-dhcp-server restart*
- *sudo systemctl status isc-dhcp-server -l*

Para concluir as configurações de conexão com o provedor via discagem *PPPoE*, antes de alterar as configurações do arquivo */network /interfaces,* foi necessário desativar e desabilitar o serviço *dhcpcd* da placa *raspberry* a fim de evitar conflitos de *IP's,* tendo em vista que o provedor da rede local realiza disponibiliza *IP's* dinamicamente. Foram utilizados os seguintes comandos para esta execução:

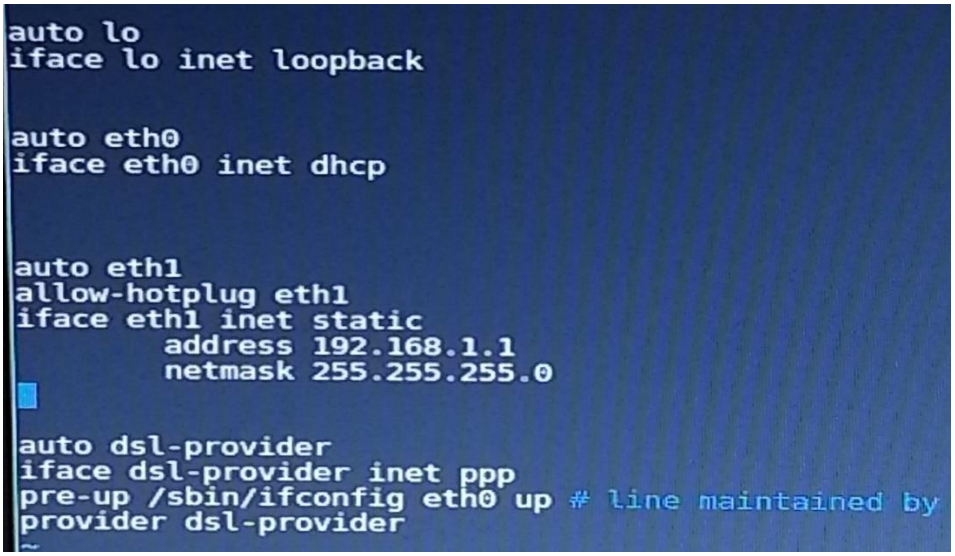
- *sudo service dhcpcd stop*
- *sudo systemctl disable dhcpcd*

Finalizando as configurações para efetivar a conexão com o provedor*,* alterações foram realizadas utilizando um editor de texto no arquivo */etc /network /interfaces.* Nesse arquivo foram configuradas a interface *loopback* (*lo*) que é responsável pela comunicação entre as máquinas da rede local, a interface *eth0,* por onde chega a *internet* e a interface *eth1* que distribui a *internet* para todas as máquinas da rede local, além da interface *ppp* que faz a conexão com o provedor. Para tais configurações foi executado o seguinte comando:

- *sudo vim /etc /network /interfaces*

Após as configurações, o arquivo */etc /network /interfaces* ficou conforme a figura 17 abaixo:

Figura 17 – Configuração do arquivo *Network Interfaces*



Fonte: Do próprio autor.

Finalmente, para estabelecer definitivamente a conexão PPPoE entre o provedor e a rede local e verificar o status, foram executadas as seguintes linhas de comando abaixo:

- sudo pon dsl-provider
- sudo plog

Estabelecida a conexão com o provedor, para a internet poder chegar a todas as maquinas da rede local, foi necessário configurar o roteador no modo *Access Point* (ponto de acesso).

Inicialmente, foi preciso abrir um navegador e digitar o seguinte endereço abaixo:

- 192.168.0.1

Posteriormente, apareceu a seguinte tela, de acordo com a figura 18, onde foi digitado o nome do usuário e apertar a tecla *enter*, conforme figura abaixo:

Figura 18 – Acessando as configurações do roteador

Fonte: Do próprio autor.

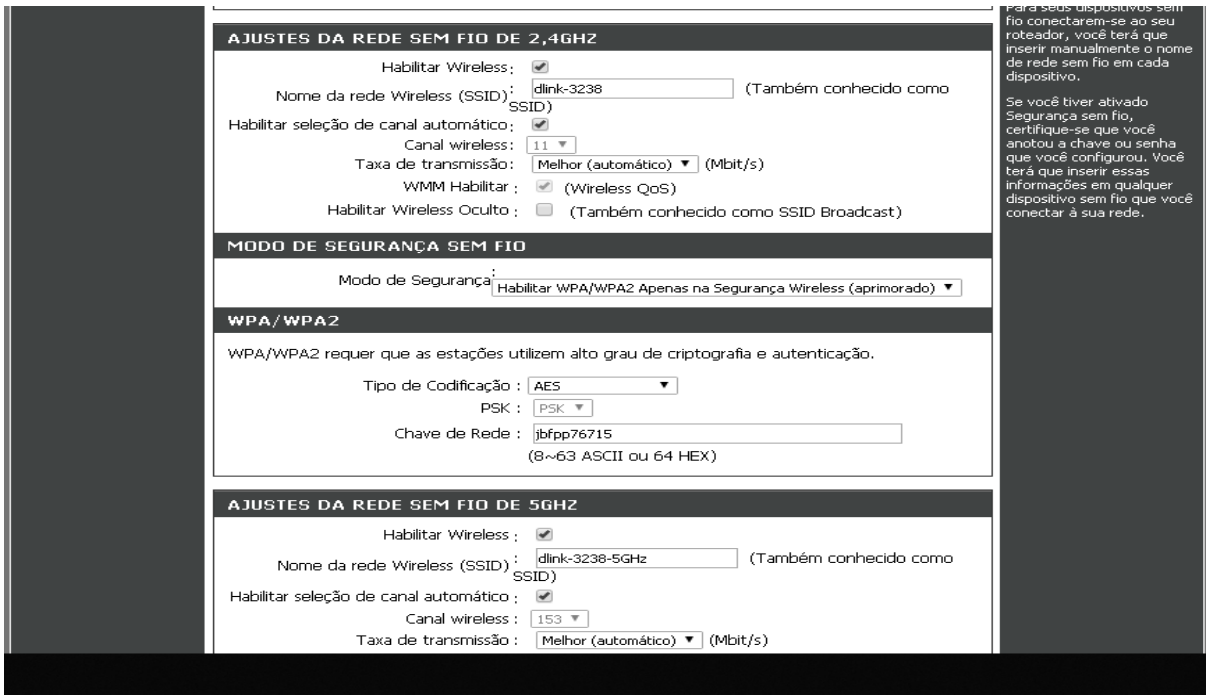
Próximo passo foi clicar na aba configuração e logo após em conexão *wi-fi*, alterando o modo *wireless* para o modo *AP*, conforme figura 19:

Figura 19 – Configurando modo *AP*

Fonte: Do próprio autor.

Em seguida foram realizados ajustes na rede sem fio, alterando o nome da rede *wireless*, desabilitando a seleção automática de canais e modificando para o canal onze (recomendado), selecionado o modo aprimorado de segurança, definido o tipo de codificação para *AES* e criando nova senha de rede, conforme figura 20 abaixo:

Figura 20 – Configuração da rede sem fio e do tipo de segurança



Fonte: Do próprio autor.

A partir dessas configurações, a comunicação do servidor *raspberry* com o roteador foi realizada utilizando a porta LAN.

Estabelecida a conexão e configurado o roteador no modo *Access Point*, todas as máquinas da rede local passaram a receber internet dentro da faixa de *IP* definida na interface *eth1*.

Dando continuidade à implementação do sistema de controle de acesso à *internet* utilizando o *raspberry pi,* foi implementado um filtro utilizando o *Proxy Squid* para realizar o controle de pacotes que chegam na rede local. O *Squid* é um *software* livre e com suporte para *Windows* e *Linux.* Sua principal função está no uso em servidores *proxy* capazes de suportar *HTTP, HTTPS, FTP,* dentre outros formatos.

Para a implementação do filtro de controle do servidor *raspberry,* utilizando o *shell*, foi necessário baixar os pacotes da ferramenta *Squid3* conforme linhas de comando abaixo:

- *sudo apt update*
- *sudo apt install squid3*

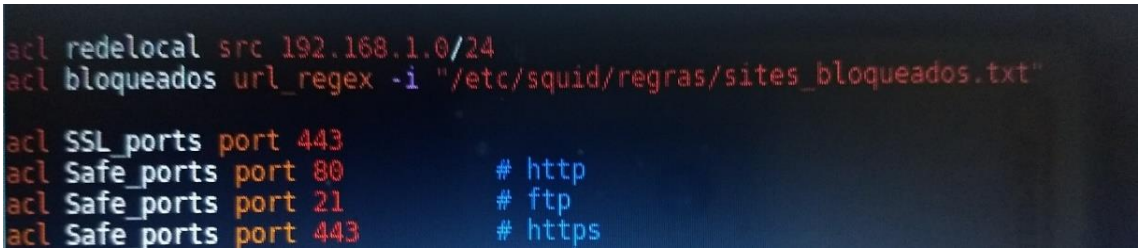
O *Squid* tem como porta padrão a 3128, a ideia é forçar todo tráfego da rede local a passar pela porta do *Squid.* Desta forma, utilizando um editor de texto no *shell,* foi realizada algumas alterações no arquivo de configuração do *Squid* utilizando a seguinte linha de comando:

- *sudo /etc /squid /squid.conf*

Acessado o arquivo de configurações, na parte das *ACL's,* listas que possuem regras de bloqueio ou liberação de acesso, foi criada uma *ACL* para a rede local a fim de cadastrar um arquivo contendo regras de acesso utilizando a seguinte linha de comando abaixo, como mostrado na figura 21:

- *acl redelocal src 192.168.1.0/ 24*

Figura 21 – Criação de *ACL* para rede local

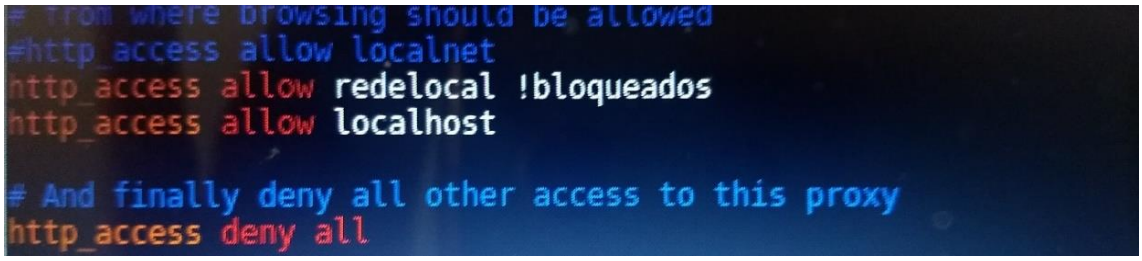


Fonte: Do próprio autor.

Em seguida, conforme figura 22, foi executada uma configuração cadastrando a rede local para que a mesma possa navegar utilizando o *Squid* como *proxy*, usando a seguinte linha de comando na sessão *http_access*:

- *http_access allow redelocal*

Figura 22 – Cadastro de *ACL* para rede local utilizar o *Squid*

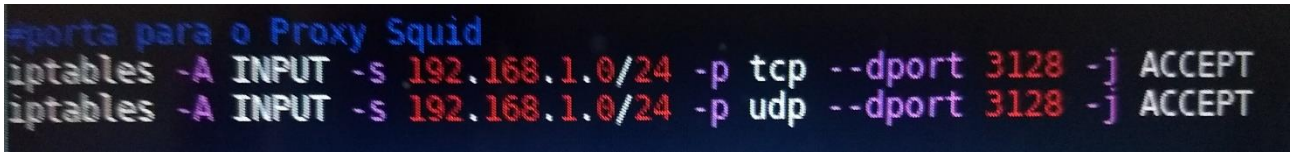


Fonte: Do próprio autor.

- *sudo /etc /init.d /squid restart*

Foi cadastrado regras no rc.local (firewall) para permitir que o squid aceite conexões através de sua porta padrão, utilizando as seguintes linhas de comando abaixo, conforme figura 23:

Figura 23 – Implementação de regras para o *Squid* no *rc.local*



Fonte: Do próprio autor.

Próximo passo foi criar um arquivo de texto dentro do diretório do *Squid*, onde ficarão armazenas as palavras, sites e domínios que se deseja bloquear. Utilizando-se do *shell* com um editor de texto foram digitadas as seguintes linhas de comando para criar o referido arquivo e dentro deste outro arquivo em branco contendo o conteúdo a ser bloqueado:

- *sudo mkdir /etc /squid /regras*
- *sudo vim /etc /squid /regras /sites_bloqueados.txt*

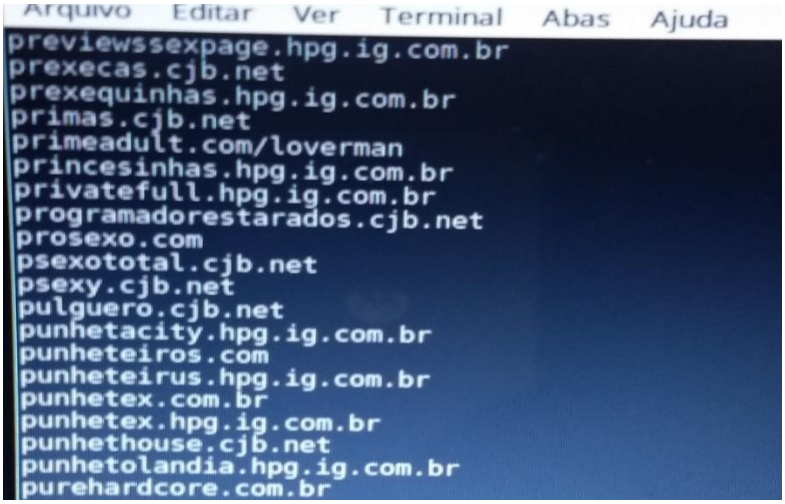
Para a implementação do conteúdo a ser bloqueado, foi consultado e baixado na *internet* arquivos contendo uma *blacklist* para a realização de testes com servidor *raspberry*. Foi utilizada uma linha de comando para copiar o arquivo *blacklist* da *internet* e este foi armazenado na pasta /home /pi.

- *Wget http//: URL_do_site/raw /arquivo_blacklist*

Em seguida, o arquivo *blacklist*, mostrado na figura 24, foi copiado para a pasta destino */etc /squid /regras /sites_bloqueados.txt* para possibilitar proceder à execução do controle de acesso, utilizando a seguinte linha de comando no *shell:*

- *cat arquivo_blacklist > /etc /squid /regras /sites_bloqueados.txt*

Figura 24 – *Blacklist* gerada para bloqueio



Fonte: Do próprio autor.

Acessando o arquivo */etc /squid /squid.conf,* foi cadastrado nas regras *ACL* o arquivo */etc /squid /regras /sites_bloqueados*, contendo o conteúdo a ser bloqueado. Para essa execução foi digitada no *shell* a seguinte linha:

- *acl bloqueados url_regex -i "/etc/ squid/ regras/ sites_bloqueados.txt*

Continuando com as configurações, no arquivo */etc /squid /squid.conf,* na sessão do *http_access* foi acrescentada uma alteração na linha *http_access allow redelocal* para que todo permitir apenas navegar se os pacotes forem diferentes do conteúdo do arquivo *blacklist,* conforme linha de comando abaixo:

- *http_access allow redelocal !bloqueados*

Realizados os procedimentos anteriores, foi necessário reiniciar o *Squid* segundo linha de comando abaixo para o filtro começar a funcionar:

- *sudo / etc/ init.d /squid restart*

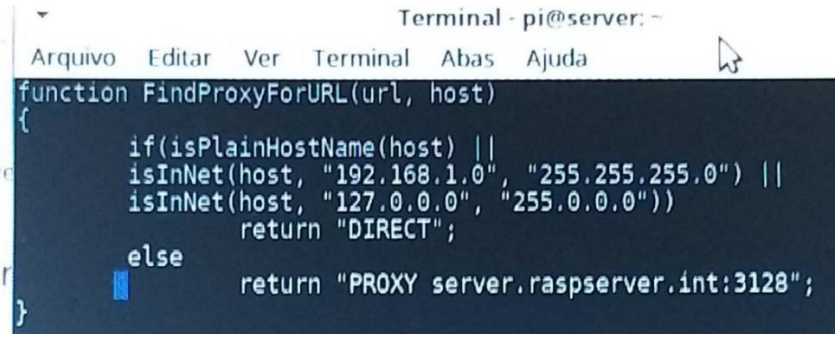
Próximo passo foi configurar o *Squid* para trabalhar com o *scrit Wpad* a fim de autoconfigurar os navegadores das máquinas da rede local, desta forma, autodetectando automaticamente as configurações de *proxy,* fazendo a detecção via rede. O método utilizado ocorrerá via *DNS*. O *Wpad* não funciona automaticamente por questões de segurança em dispositivos móveis, necessitando realizar as configurações manualmente. Para executar a autoconfiguração, foi preciso instalar a ferramenta *Apache2* conforme linha de comando abaixo e reiniciar o sistema para que o *Apache* possa ser executado.

- *sudo apt update*
- *sudo apt install apache2*

Seguindo com a implementação, foi criado na pasta */var /www /html* um arquivo *Wpad*, conforme figura 25, contendo o *script* de autoconfiguração dos navegadores, segundo a linha de comando a seguir:

- *sudo vim /var /www /html /wpad.dat*

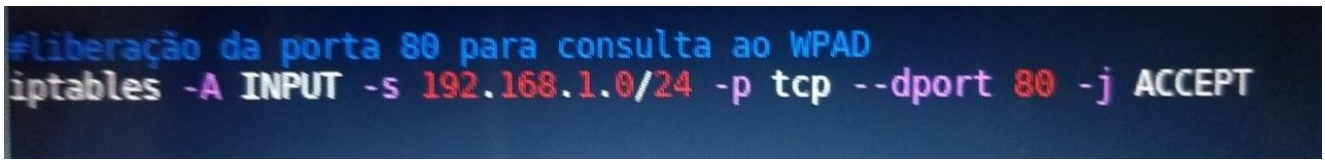
Figura 25 – *Script* de autoconfiguração dos navegadores



Fonte: Do próprio autor.

Finalizando a parte do *scrip wpad,* foi acrescentado no arquivo */etc /rc.local* (*firewall*) mais uma regra para liberar acesso do navegador na porta 80 via *http*, como mostrado na figura 26..

Figura 26 – Implementação no arquivo *rc.local* para o *script wpad*



Fonte: Do próprio autor.

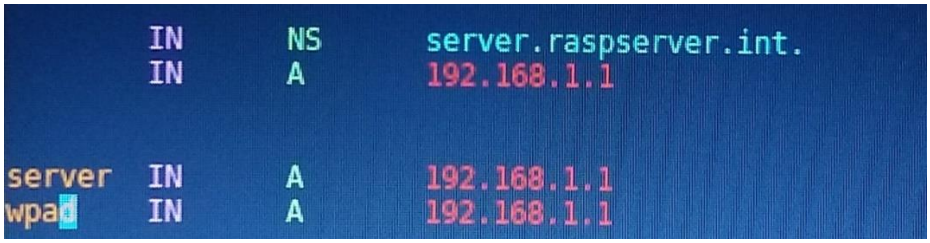
Para que a nova regra pudesse ser executado foi necessário reiniciar o arquivo */etc /rc.local.*

- *sudo /etc /rc.local start*

A fim de tornar funcional o sistema de controle de acesso à internet, foram realizados procedimentos de autoconfiguração dos navegadores com *Wpad* de forma automática. Para isso foi preciso incluir o registro *wpad* no arquivo *DNS* para que seja encontrado no domínio *raspserver.int* o *wpad.raspserver.int*, tendo na raiz o arquivo *wpad.data.*

Acessando os arquivos da zona local */etc /bind /raspserver.int.lan,* foi inserido um novo registro *wpad* com o seu respectivo *IP,* conforme figura 27 a seguir:

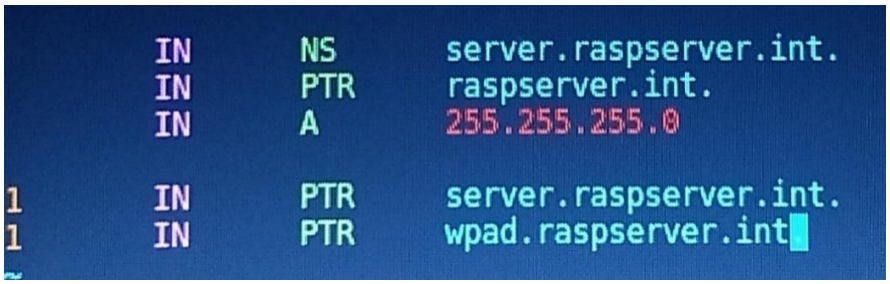
Figura 27 – Registro do *script wpad* no arquivo *raspserver.int.lan*



Fonte: Do próprio autor.

O mesmo procedimento foi executado no arquivo de *DNS* reverso, o */etc /bind /1.168.192.db,* porém, inserindo o número final do *IP* com o seu respectivo domínio, segundo figura 28.

Figura 28 – Registro do *script wpad* no arquivo reverso



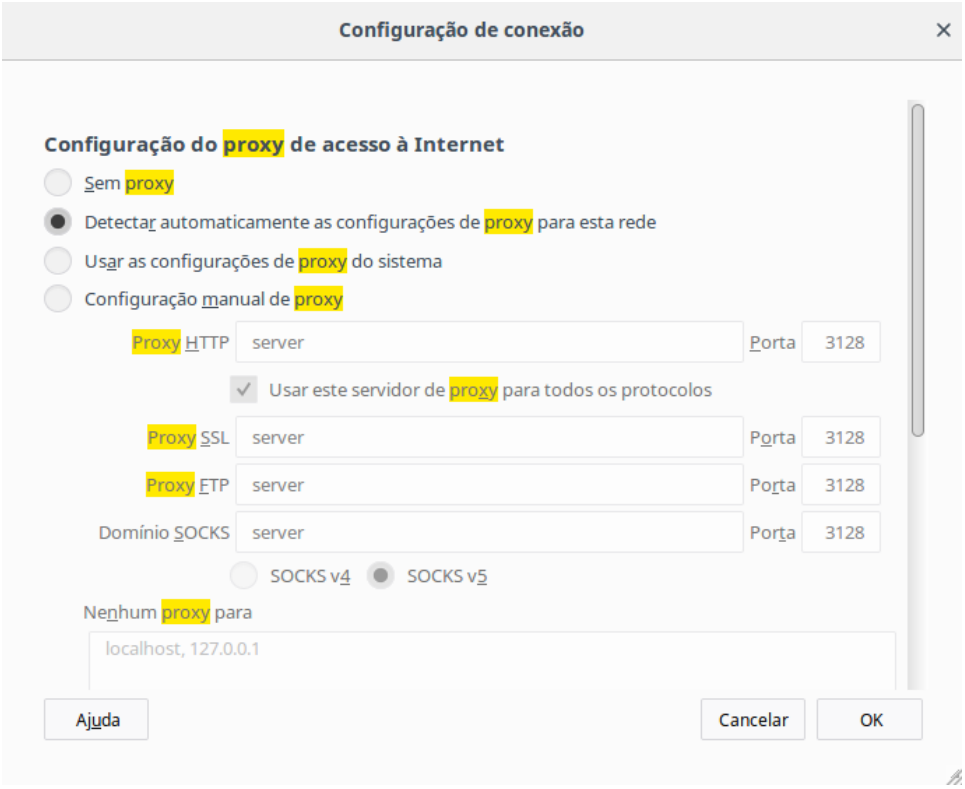
Fonte: Do próprio autor.

Após as configurações nos arquivos *DNS*, o serviço *bind* foi reiniciado.

- *sudo service bind9 restart*

Para finalizar a autoconfiguração dos navegadores, foi acessada a opção “configurações” do navegador, buscando por configurações de *proxy*, em seguida foi escolhida a opção “configurações de *LAN,* deixando marcada apenas a opção de detecção automática de *proxy*, segundo figura 29.

Figura 29 – Ajuste para detecção automática do *proxy*



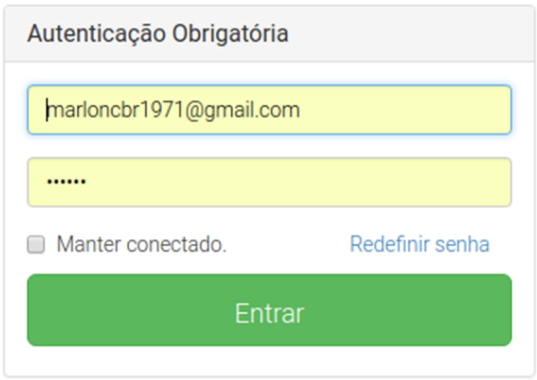
Fonte: Do próprio autor.

2.1 Integração de Interface Gráfica

Para potencializar o sistema residencial de controle de acesso à *internet* em questão, foi realizada a integração de um produto de monografia, cujo autor, Marlon Francisco dos Santos, apresentou a banca nas Faculdades Unificadas Doctum de Teófilo Otoni no ano de 2017, com o tema: Implementação de painel de administração para gerenciamento do *Squid*.

A interface gráfica é simples e de fácil utilização, apresentando uma tela para realizar o login (usuário e senha) conforme a figura 30 a seguir:

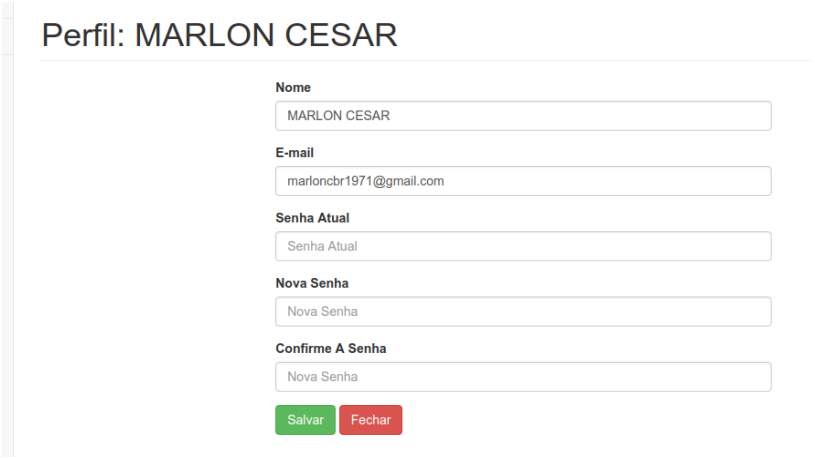
Figura 30 – Aparência da tela de *login*



Fonte: Do próprio autor

A interface gráfica também possui uma tela para cadastrar usuário conforme a figura 31:

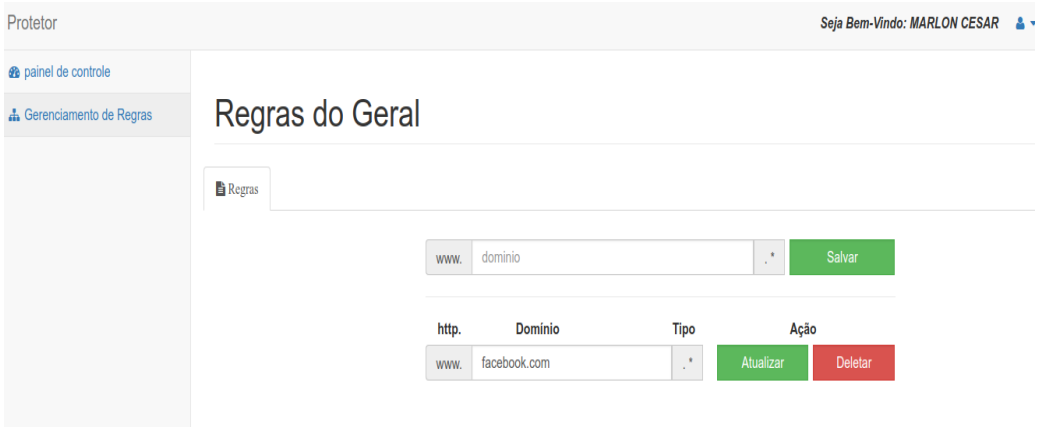
Figura 31 – Tela para cadastro de usuário



Fonte: Do próprio autor

A seguir, segundo a figura 32, é apresentada a tela para gerenciamento das regras de bloqueio:

Figura 32 – Tela para gerenciamento do controle de acesso



Fonte: Do próprio autor

Para acessar o sistema é preciso digitar o endereço no navegador como mostrado na figura 33:

Figura 33 – Endereço para acessar o sistema

Fonte: Do próprio autor

Finalizadas as configurações, foram realizados testes práticos em laboratório e constatada a funcionalidade do sistema de controle de acesso à internet do *raspberry pi*.

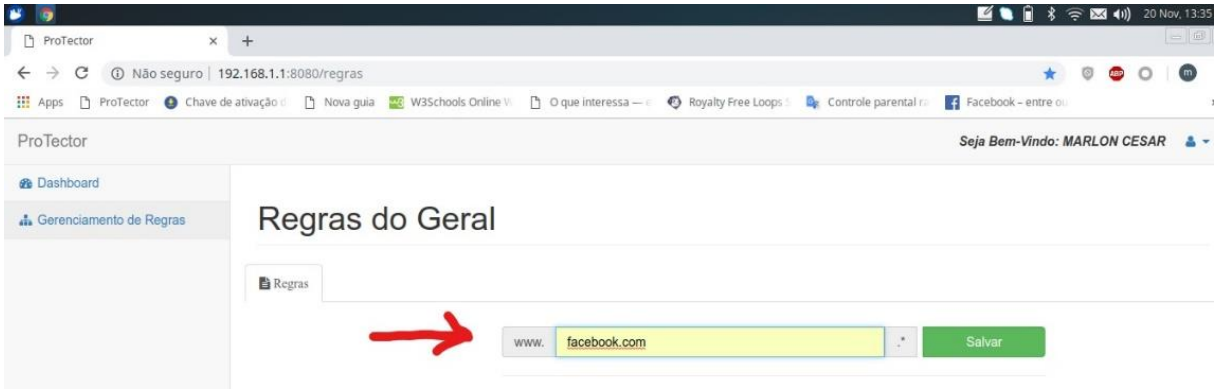
3 TESTES

Após a implementação do sistema com as devidas configurações e integração da interface gráfica, foram realizados testes a fim de verificar a funcionalidade e eficiência da ferramenta proposta.

Inicialmente, foi cadastrada uma regra (facebook) a ser bloqueada pelo sistema conforme a figura 34 abaixo:

.

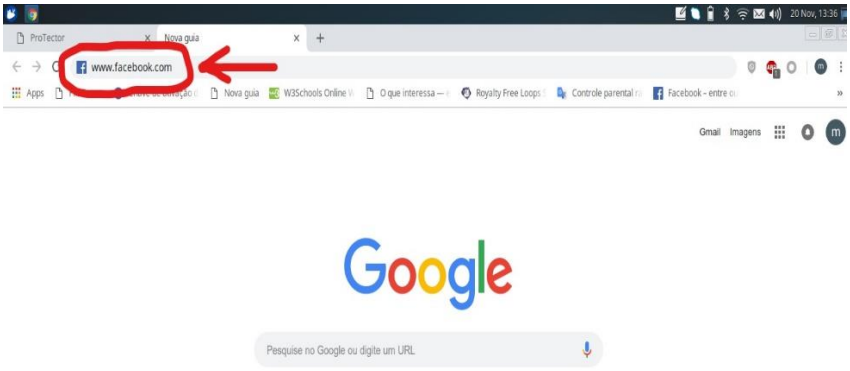
Figura 34 – Tela de seleção do site a ser bloqueado na tela de regras



Fonte: Do próprio autor

Em seguida, foi inserido em um navegador o endereço do site (facebook) a bloqueado pelo sistema, segundo figura 35 a seguir:

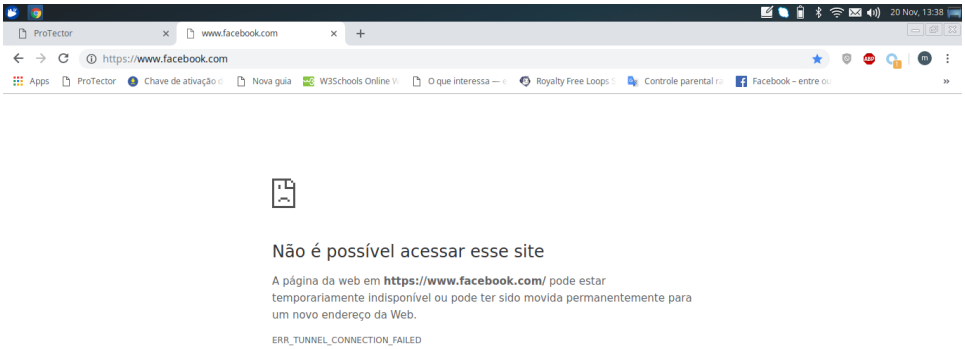
Figura 35 – Tela de busca pelo site no navegador



Fonte: Do próprio autor

De acordo com a figura 36 abaixo, após a tentativa de acessar a página a ser bloqueada, foi constatada a efetividade do bloqueio do site:

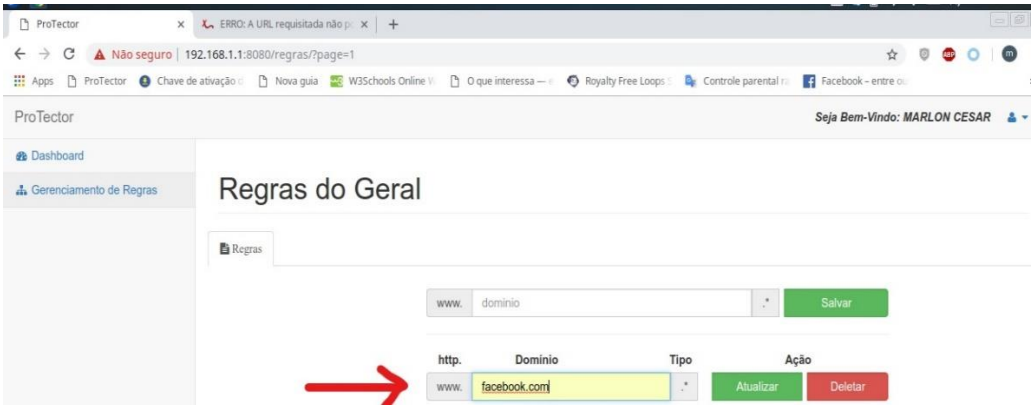
Figura 36 – Tela de bloqueio realizado com sucesso



Fonte: Do próprio autor

Efetivado o bloqueio, o próximo passo foi realizar a exclusão da regra conforme a figura 37 abaixo:

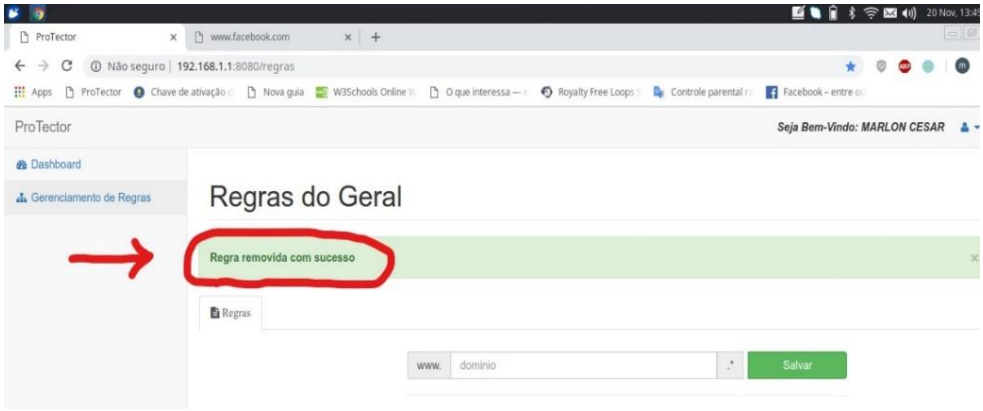
Figura 37 – Tela de desbloqueio do site



Fonte: Do próprio autor

Após a exclusão, o sistema mostra na tela a confirmação da regra deletada como pode ser visto na figura 38 abaixo:

Figura 38 – Tela confirmação de desbloqueio



Fonte: Do próprio autor

Segundo a figura 39 abaixo, após a confirmação da exclusão da regra de bloqueio, o site pode ser acessado:

Figura 39 – Tela de desbloqueio realizado com sucesso

Fonte: Do próprio autor

3.1 Análise dos Testes

Após a realização de testes, foi constatado êxito nos objetivos propostos pelo presente trabalho, visto que, utilizando-se de um microcomputador e de um sistema operacional *raspbian*, específico para o *raspberry pi* e, realizando a instalação e configuração de ferramentas como o Apache, *squid*, *DNS*, *DHCP*, foi possível a implementação de um sistema capaz de realizar o controle de acesso à *internet*, além de oferecer como uma ferramenta de baixo custo a *raspberry pi*, quando comparada a dispositivos similares no mercado, conforme pesquisa realizada em sites de compra na internet, estabelecendo uma relação de custo-benefício como mostrado no quadro 01 abaixo:

Quadro 01: Preços

DISPOSITIVOS	PREÇOS
Raspberry PI	U$35,0
Rock 64	U$25,0
Hickey 960	U$239,0
Tinker Board	U$60,0

Fonte: Do próprio autor

Apesar de apresentar inicialmente um preço inferior, a desvantagem da *Rock 64* em relação à placa *raspberry pi* se encontra na indisponibilidade de serviços como bluetooth e Wireless, sendo necessária a aquisição de dispositivos auxiliares, desta forma, tornando o custo da referida placa mais alto.

CONCLUSÃO

Após intensas pesquisas e trabalhos de configuração realizados para a implementação de um sistema residencial de controle de acesso à internet utilizando o raspberry pi 3 model b, conclui-se que os resultados obtidos para o presente trabalho se apresentaram eficientes e de baixo custo em relação a outros produtos no mercado. Tendo em vista que a proposta é bloquear o máximo possível de sites inapropriados para crianças e adolescentes, testes práticos realizados através de simulações de acesso a sites se demonstraram eficazes utilizando o proxy squid como filtro.

Algumas das ferramentas utilizadas foram: O pacote *pppoeconf*, o *Squid*, *Bind9*, *DHCP*, *Shell Script*, *Firewall*, *Raspbian*, o apache e uma interface integrada, sendo todas, essenciais para a implementação e configuração da ferramenta proposta, uma vez que possibilitaram a comunicação do *raspberry* com a rede externa e as maquinas da rede local, a resolução de nomes, a implementação de regras para bloqueio e acesso à internet, serviços de distribuição dinâmica de *IP's*, bem como a utilização do sistema pelo usuário administrador.

Como objetivo geral, o sistema de controle de acesso à *internet* utilizando o *raspberry pi* se apresentou eficiente no que diz respeito à controle de conteúdo, alcançando o objetivo ao qual foi proposto neste trabalho.

Quanto aos objetivos específicos, foram alcançados resultados satisfatórios, respondendo à questão de viabilidade das seguintes hipóteses levantadas neste trabalho:

H0 – A implementação de um sistema residencial de controle de acesso à internet utilizando o raspberry PI 3 não seria viável, considerando-se o custo-benefício da aplicação;

O sistema residencial de controle de acesso à internet proposto neste trabalho dispensa a utilização de máquinas robustas como o *desktop*, consequentemente, reduzindo os custos significativamente, tornando nula a hipótese H0.

H1 – A utilização do *raspberry pi* 3 para implementação de um sistema residencial de controle de acesso à internet seria viável, tendo em vista a possibilidade de se utilizar o sistema *Linux Raspbian* que possui ferramentas de acesso à *internet*, configurações de rede wireless, bem como excelente capacidade para uso do tipo doméstico;

O *raspberry pi* obteve um bom desempenho utilizando o sistema operacional *raspbian* que auxiliou na implementação do sistema, possibilitando a conexão com a *interne*t, validando a hipótese H1.

H2 – A implementação de um sistema residencial de controle de acesso à internet usando o *raspberry pi* 3 seria viável, visto que se pode configurá-lo como um servidor, utilizando serviços para prover segurança à rede;

Neste trabalho foi possível a utilização de ferramentas como Apache, *DNS, DHCP* e *Squid* para a configuração do servidor, possibilitando a implementação do sistema de controle de acesso à *internet*, desta forma, validando a hipótese H2.

H3 – A implementação de um sistema residencial de controle de acesso à internet utilizando o *raspberry pi* seria viável, considerando-se que possivelmente necessitaria de outro *hardware* como um switch para possibilitar a implantação do sistema.

Além do roteador que foi utilizado para o compartilhamento da *internet* com as máquinas da rede local, na parte de *hardware*, foram necessários apenas uma placa *raspberry pi* 3 modelo B e um adaptador de rede RJ45 com entrada *HDMI*, dispensando o uso de qualquer outro equipamento como *switch* para a implementação do sistema, validando a hipótese H3.

H4 – A utilização do *raspberry pi* 3 como eficaz alternativa para implementação de um sistema residencial de controle de acesso à *internet* seria viável, considerando-se a relação custo-benefício em comparação com os *desktops* e dispositivos similares ao *raspberry* como o *Rock* 64, *Hickey* 960 e a *Tinker Board*.;

De acordo com pesquisas de mercado realizada em sites de compra na *internet*, ficou constatada a vantagem do *raspberry pi* na relação custo-benefício comparando-se com equipamentos similares, validando com isso a hipótese H4.

O sistema residencial de controle de acesso à *internet* se mostrou bastante funcional, tendo em vista que se utilizando de um *script wpad* foi possível realizar a autoconfiguração dos navegadores para que possam detectar automaticamente as configurações do *proxy*.

Apesar dos testes práticos na rede local do laboratório terem obtido êxito, não houve tempo hábil para sua implantação em um ambiente real. Para fins de continuidade, apresento sugestões de melhorias ao trabalho proposto, como a aplicação de controle parental, controle de acesso através do *IP*, utilização do dispositivo *PoE Hat,* que adaptado à placa *raspberry* possibilita alimentação elétrica usando o próprio cabo de rede juntamente com os dados, além de configuração da placa *raspberry* para trabalhar em modo *wirelless,* dispensando possivelmente o uso de roteadores, desta forma, reduzindo o custo de implantação e utilização da ferramenta.

REFERÊNCIAS

BRESNAHAN, Christine; NEGUS, Christopher. *Linux:* A Bíblia. 8ed. Rio de Janeiro: Alta Books, 2014.

MOTA FILHO, João Eriberto. *Análise de Tráfego em Redes TCP/IP*. 1ed. São Paulo: Novatec, 2013.

NEVES, Julio C. *Programação Shell Script*. 11ed. Rio de Janeiro: Brasport, 2017.

SANTOS, Márlon Francisco dos. *Implementação de Painel de Administração para Gerenciamento do Squid*. 2017. 50 f. Monografia (Graduação em Sistemas de Informação) – Faculdades Unificadas Doctum de Teófilo Otoni, Minas Gerais, 2017.

TANENBAUM, Andrew S. *Redes de Computadores*. 4ed. Rio de Janeiro: Elsevier, 2003.

VASCONCELOS, Laércio; VASCONCELOS, Marcelo. *Manual Prático de Redes*: Aprenda Redes Pelo Lado Prático. 1ed. Rio de Janeiro: Laércio Vasconcelos Computação, 2008.

Disponível em: <https://www.raspberrypi.org/about/>. Acesso em: 05 mai. 2018.

Disponível em: <https://www.raspberrypi.org/products/poe-hat/>. Acesso em: 5 mai. 2018.

Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 10 mai. 2018.

Disponível em: <https://www.squid-cache.org/intro/>. Acesso em: 10 mai. 2018.

Disponível em: <https://www.gnu.org/philosophy/free-sw.pt-br.html>. Acesso em: 10 mai. 2018.

Disponível em: < https://www.gnu.org/>. Acesso em: 13 mai. 2018.

Disponível em: < https://www.raspbian.org/RaspbianAbout>. Acesso em 16 mai. 2018.

Disponível em:<http://painel.passofundo.ifsul.edu.br/uploads/arq/20160711175902455087226.pdf>. Acesso em 16 mai. 2018.

Disponível em: <https://www.isc.org/downloads/dhcp/>. Acesso em 17 mai. 2018.

Disponível em: <https://canaltech.com.br/internet/o-que-e-dns/>. Acesso em 18 mai. 2018.

Disponível em: <https://www.apache.org/>. Acesso em 20 mai. 2018.