

INSTITUTO FEDERAL

Sul de Minas Gerais

Campus Inconfidentes

LEANDRO LUIS DE MORAES

MATHEUS RODRIGUES DA SILVA

**DESENVOLVIMENTO DE UMA APLICAÇÃO PARA
INTEGRAR O PHPIPAM COM O ISCDHCP**

INCONFIDENTES - MG

2017

LEANDRO LUIS DE MORAES
MATHEUS RODRIGUES DA SILVA

**DESENVOLVIMENTO DE UMA APLICAÇÃO PARA
INTEGRAR O Phipam COM O ISCDHCP**

Trabalho de Conclusão de Curso apresentado como pré-requisito de conclusão do curso de Graduação Tecnológica em Redes de Computadores no Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Inconfidentes, para obtenção do título de Tecnólogo em Redes de Computadores.

Orientador: Kleber Marcelo da Silva Rezende

INCONFIDENTES - MG

2017

LEANDRO LUIS DE MORAES
MATHEUS RODRIGUES DA SILVA

**DESENVOLVIMENTO DE UMA APLICAÇÃO PARA
INTEGRAR O PPIPAM COM O ISCDHCP**

**Data de aprovação:
30 de Outubro de 2017**

Orientador: Prof. Kleber Marcelo da Silva Rezende
IFSULDEMINAS – Campus Inconfidentes

Prof. Vinícius Ferreira de Souza
IFSULDEMINAS – Campus Inconfidentes

Prof. Igor Oliveira Lara
IFSULDEMINAS – Campus Inconfidentes



Instituto Federal de Educação, Ciência e
Tecnologia
Sul de Minas Gerais
Campus Inconfidentes



Praça Tiradentes, 416 - Centro - CEP 37576-000
Telefone: (35) 3464-1200

Desenvolvimento de uma aplicação para integrar o phpIPAM com o ISCDHCP

Matheus R. da Silva¹, Leandro L. de Moraes¹, Kleber M. da S. Rezende¹

¹Setor de Informática e Redes

IFSULDEMINAS – Campus Inconfidentes – Inconfidentes, MG – Brasil

matheursilva@gmail.com, leandrollmoraes@gmail.com,
kleber.rezende@ifsuldeminas.edu.br

Abstract. Nowadays, the computer network management has been facing new challenges. One of the biggest problems is mapping the large amount of IP addresses of all devices connected to the network. IPAM tools help in this work of accounting and organizing the addresses, but fail to have any real control over the network. The target application of this work aims to create an interface to integrate phpIPAM with a Linux DHCP server using PHP in order to control the network addressing. The solution accesses the phpIPAM database, displays the registered networks and allows them to be selected to export. The application generates an ISC-DHCP configuration file, which is later sent to the DHCP server via SSH. The results obtained have been satisfactory, allowing the expansion of the application functionalities, in order to facilitate, even more, the network configuration tasks.

Resumo. Atualmente a administração de redes de computadores vem enfrentando novos desafios. Um dos maiores problemas é mapear a grande quantidade de endereços IP de todos os dispositivos conectados à rede. Ferramentas IPAM auxiliam nesse trabalho de contabilizar e organizar os endereços, mas falham por não possuir nenhum tipo de controle real sobre a rede. A aplicação alvo desse trabalho tem como objetivo criar uma interface para integrar o phpIPAM com um servidor DHCP Linux, utilizando PHP, a fim de controlar o endereçamento da rede. Tal solução acessa a base de dados do phpIPAM, exibe as redes registradas e permite que elas sejam selecionadas para a exportação. A aplicação gera um arquivo de configuração do ISC-DHCP, que é, mais tarde, enviado ao servidor DHCP via SSH. Os resultados obtidos foram satisfatórios, possibilitando a expansão das funcionalidades da aplicação, de forma a facilitar, ainda mais, as tarefas de configuração da rede.

1. Introdução

Devido à atual complexidade das redes de computadores, que hoje englobam não apenas computadores de usuários mas também sofrem com as tendências de BYOD (*Bring Your Own Device*) e IoT (*Internet of Things*), acrescentando dispositivos como câmeras, telefones VoIP, sensores, celulares e notebooks pessoais, um dos grandes problemas dos seus administradores é mapear os endereços IP dos dispositivos que a compõem. Esse mapeamento é de extrema necessidade, visto o aumento de acesso remoto, utilizando o endereço IP dos dispositivos, como programas de assistência remota ao usuário ou gerência de algum equipamento de IoT da rede por exemplo. Considerando esta demanda crescente por aplicações que auxiliem nesta tarefa de mapeamento da rede, nasceram as aplicações IPAM (*IP Address Management*) que, embora sejam úteis na organização dos endereços IP, não podem ser consideradas ferramentas de gerência, visto que não atuam diretamente na rede. Ferramentas IPAM são capazes de descobrir dispositivos na rede, mas não são capazes por si só de corrigir ou alterar algo nela. Apesar desta limitação, uma ferramenta se destacou: o phpIPAM (PHPIPAM, 2017), aplicação criada em PHP (*PHP Hypertext Processor*), sendo uma das mais completas em termos de usabilidade e organização atualmente.

Considerando tais fatores, observa-se uma possibilidade de utilizar a base de dados do phpIPAM para a configuração de um servidor DHCP (*Dynamic Host Configuration Protocol*) (ARAUJO, 1997), possibilitando o controle da rede a partir de um software IPAM. Uma configuração interessante de um servidor DHCP para esta função é a atribuição estática de endereços, sendo possível configurar cada endereço IP associando à um MAC específico diretamente no servidor, dispensando a necessidade da locomoção até o local desses dispositivos, como ocorreria caso fosse utilizado o endereçamento IP manual.

Diante do exposto, o presente trabalho buscou desenvolver uma solução livre, que possibilitasse a integração destas ferramentas.

2. Materiais e Métodos

A solução se baseia em servidores baseados em Linux, visto que, de acordo com um estudo recente, 83% dos entrevistados estão rodando Linux em seus servidores (SUSE, 2013).

Para o desenvolvimento, foi selecionada a linguagem PHP, que é a linguagem nativa do phpIPAM e, além disso, utilizada em 82,7 % dos servidores web no mundo (W3TECHS, 2017a).

A topologia de testes foi desenvolvida em um ambiente virtual emulado, dada a inviabilidade do uso de equipamentos reais para implementação do projeto, utilizando o simulador de redes GNS3 (*Graphical Network Simulator 3*). O GNS3 permite a criação de redes de qualquer tamanho e topologia e emula os mais diversos equipamentos de redes de todos os fabricantes, além de permitir a inclusão de máquinas virtuais e o próprio computador do usuário no projeto. A figura 1 apresenta a topologia utilizada para o desenvolvimento e testes da aplicação.

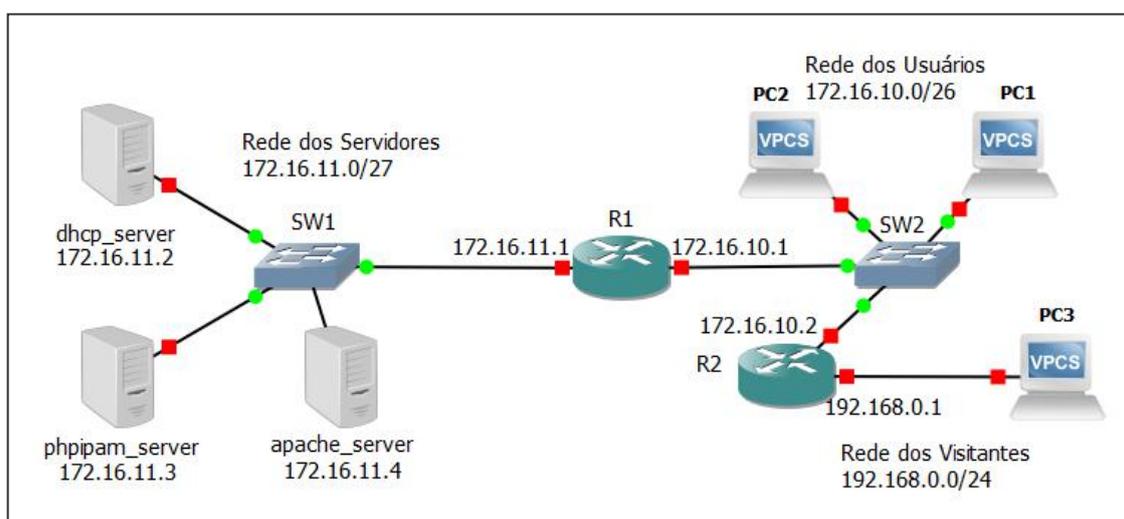


Figura 1. Topologia de rede virtual emulada pelo software GNS3.

A topologia utilizada para o desenvolvimento faz uso de três servidores. Um servidor DHCP e dois servidores web, um para instalação do phpIPAM e outro para o software desenvolvido. Para os servidores foi escolhida a distribuição Debian, uma das distribuições mais estáveis utilizadas para servidores. Nos três servidores foi instalado o pacote *openssh-server*, para permitir o acesso remoto e a comunicação entre eles de forma segura, quando necessário, via SSH (*Secure Shell*). Para o servidor DHCP foi utilizado o pacote *isc-dhcp* (INTERNET SYSTEMS CONSORTIUM, 2017). Para os dois servidores web foi utilizado o pacote *apache*, o mais utilizado para a linguagem PHP (W3TECHS, 2017b). Além disso em um dos servidores web também foi instalado o pacote *mysql-server*, para o banco de dados do phpIPAM.

Configurando corretamente os endereços de cada um dos roteadores e servidores, os roteadores R1 e R2 foram configurados para encaminhar solicitações DHCP ao endereço do servidor DHCP, nomeado como `dhcp_server`.

A aplicação foi implementada no servidor web `apache_server` e o `phpIPAM` instalado no `phpipam_server`. Quanto às configurações dos servidores, todos foram configurados para que tivessem acesso de `root` por SSH liberado e tiveram suas chaves públicas registradas uns nos outros, permitindo que a comunicação entre eles acontecesse sem o uso de senhas. O `phpIPAM` foi configurado para que atendesse à topologia estabelecida, como mostra a figura 2.

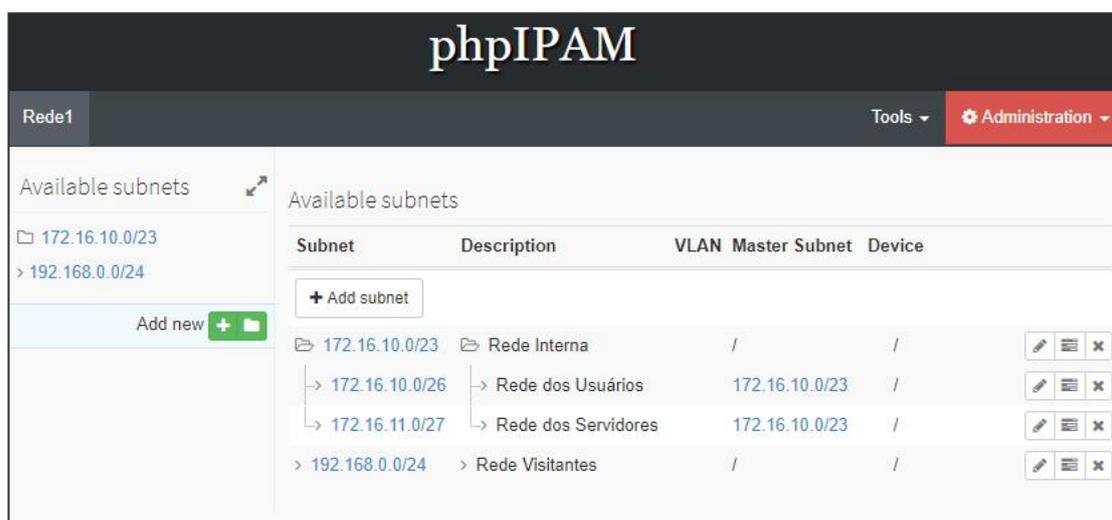


Figura 2. Registro das redes no phpIPAM.

Os dispositivos devem ser registrados no `phpIPAM` para que a ferramenta desenvolvida seja capaz de identificar o que fazer com cada um dos endereços disponíveis da rede. A tela de registro de endereços pode ser vista abaixo, na figura 3. É importante ressaltar a importância do campo *Tag*. O campo *Tag* é o campo que a ferramenta usa para definir como o endereço será utilizado. As *tags* utilizadas no projeto estão listadas na figura 4, e a explicação de como cada uma interfere na configuração será exposta mais a frente neste artigo.

Após realizar essa configuração, o administrador da rede pode editar o arquivo `config.php` para configurar os endereços e credenciais dos outros servidores. A aplicação, então, acessa a base de dados do `phpIPAM` para listar as redes disponíveis.

Figura 3. Registro de um endereço no phpIPAM.

Para visualizar e estudar o banco de dados, foi utilizada a ferramenta MySQL Workbench, que permite a melhor visualização de um banco de dados relacional. A base de dados do phpIPAM possui diversas tabelas. Nesta aplicação são utilizadas três delas: uma tabela responsável por armazenar dados referentes às redes, uma tabela para armazenar os *hosts* da rede e uma tabela para definir as *tags*. A figura 5 mostra as tabelas utilizadas pelo software.

type	Show Tag	BG color	FG color	Compress range	Locked	
Offline	Yes	#f59c99	#ffffff	No	Yes	 
Used	No	#a9c9a4	#ffffff	No	Yes	 
Reserved	Yes	#9ac0cd	#ffffff	No	Yes	 
DHCP	Yes	#c9c9c9	#ffffff	Yes	Yes	 

Figura 4. Tags utilizadas para o projeto no phpIPAM.

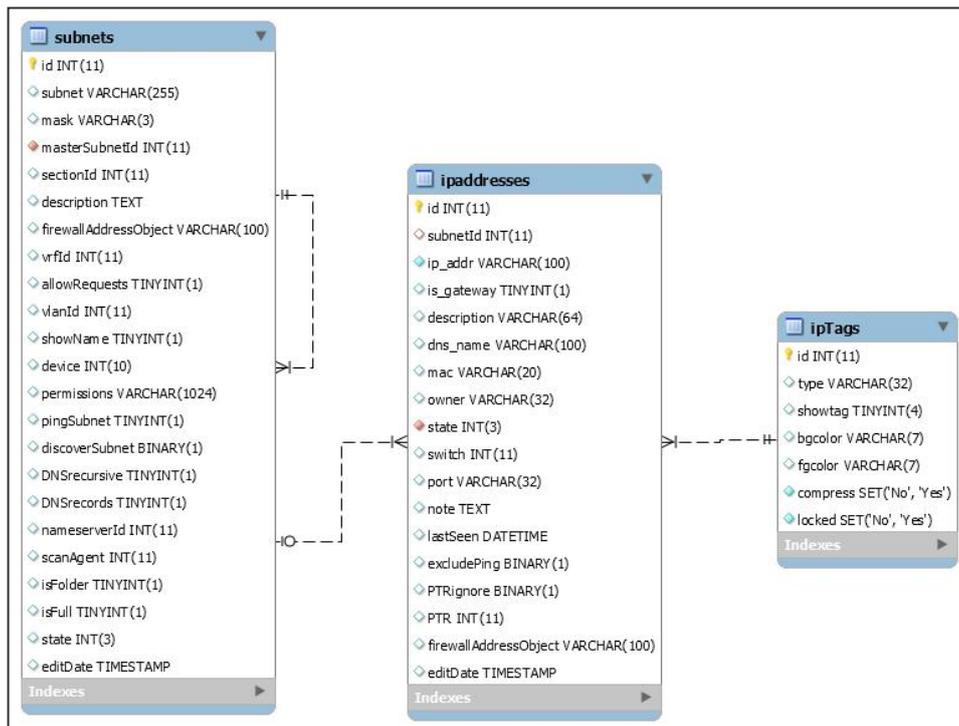


Figura 5. Tabelas utilizadas para salvar dados de redes e hosts do phpIPAM visualizada no MySQL Workbench.

A tabela *subnets* possui um relacionamento recursivo, permitindo que o software seja capaz de identificar quais são as redes registradas e suas respectivas subredes. As redes originais possuem o campo `masterSubnetId` com o valor 0. Para que o software seja capaz de listar todas as redes com suas subredes, ele também faz uso de uma função recursiva, mostrada na figura 6.

```
function getSubnets($father = 0) {
    $query = "SELECT * FROM subnets WHERE masterSubnetId = $father";
    $result = conexao()->query($query);
    $array_subnets = array();
    while ($subnet = $result->fetch_object()) {
        $network = new Subnet(
            $subnet->description,
            $subnet->subnet,
            $subnet->mask,
            getSubnets($subnet->id),
            $subnet->id
        );
        $array_subnets[] = $network;
    }
    return $array_subnets;
}
```

Figura 6. Função responsável por percorrer a base de dados do phpIPAM e gerar a árvore de redes.

A função resulta uma estrutura de dados em árvore semelhante a uma Árvore B, porém não tendo a função de ser uma árvore de busca. Cada objeto do tipo *Subnet*

possui entre os seus atributos um vetor, também do tipo *Subnet*, contendo todas as suas subredes. Com esses dados, a aplicação lista as redes de forma hierárquica na tela, assim como mostrado na figura 7.

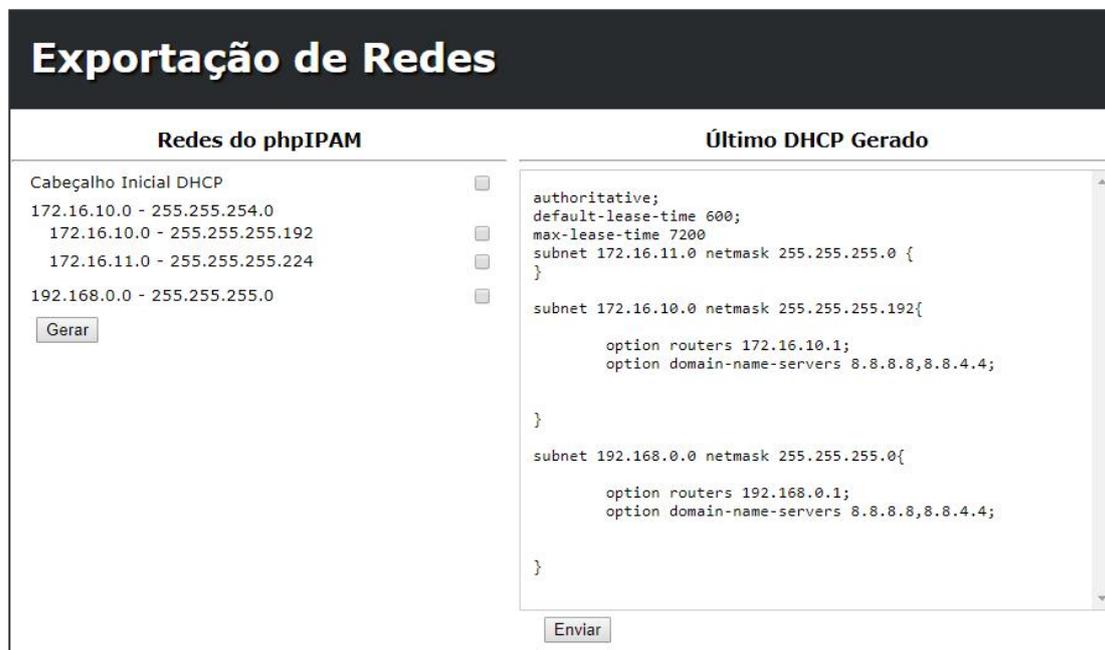


Figura 7. Tela da aplicação.

O administrador pode selecionar quais redes deseja exportar para o arquivo de configuração no lado esquerdo da aplicação. Ao selecionar as redes, a aplicação, utilizando a função *get_subnet()* percorre, novamente, o banco de dados, selecionando todos os dados da configuração de rede. Para cada uma dessas redes, é criada uma *string* que comporá o arquivo de configuração do servidor DHCP utilizando a função *printDHCPConfig()*. Essa função primeiramente cria a *string* de configuração de subrede e após isso inclui as strings criadas pelas funções *print_gateway()*, *print_dns_dhcp()*, *print_range_dhcp()* e *print_used_dhcp()* para realizar a configuração completa, explicitada na figura 8.

A função *print_gateway()* procura, na tabela *ipaddresses* algum dos endereços cadastrados que contenha o campo *is_gateway* igual a 1 e cria a *string* de configuração para que o *gateway* seja configurado.

A função *print_dns_dhcp()* usa os dados já registrados pela função *get_subnet()* e cria a *string* para que o DNS (*Domain Name Service*) seja configurado.

A função *print_range_dhcp()*, através da função *getHostsByState()* procura os endereços registrados com o campo *Tag* de valor 4, utilizado pelo phpIPAM para definir endereços disponíveis da *pool* para alocação dinâmica e cria a *string* que define esse *pool* para a configuração no servidor DHCP.

A função *print_used_dhcp()*, através da função *getHostsByState()* procura os endereços registrados com o campo *Tag* de valor 2, utilizado pelo phpIPAM para definir endereços que serão alocados estaticamente. Para cada um dos endereços, é chamada a função *print_host_dhcp()*, que cria a *string* que define o registro dessa configuração para o servidor DHCP.

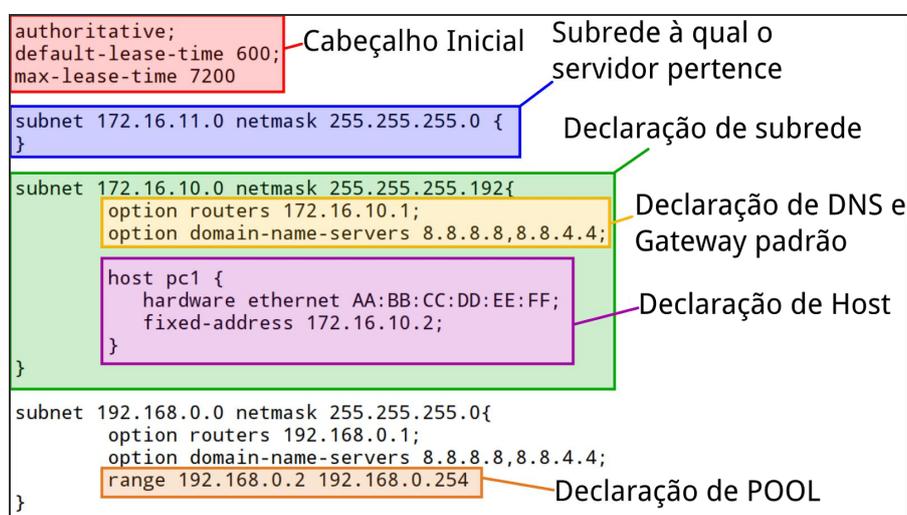


Figura 8. Esquema do arquivo /etc/dhcp/dhcpd.conf

As strings criadas são escritas, localmente, um arquivo de configuração do DHCP. Esse arquivo é exibido no lado direito da janela, vista na figura 7. Clicando no botão enviar, o arquivo é enviado via SSH ao servidor *dhcp_server*. O serviço DHCP do servidor é reiniciado e as configurações selecionadas passam a fazer efeito. Os códigos das funções especificadas estão descritos do Apêndice A.

3. RESULTADOS E DISCUSSÕES

A aplicação resultante do trabalho atendeu aos requisitos e apresentou resultados satisfatórios. A aplicação é um software de código aberto que é capaz de configurar um servidor DHCP utilizando os dados registrados no phpIPAM. O phpIPAM é responsável por manter a rede organizada e bem documentada e o DHCP permite definir o endereço de todas as máquinas da rede. Além disso a aplicação permite o uso de outras topologias,

sendo possível utilizar três servidores diferentes ou acumulando mais de uma função por servidor.

Vale acrescentar que, apesar do objetivo do projeto ser a automatização da configuração do DHCP utilizando os dados do phpIPAM, nada impede que o administrador de rede faça alterações nas configurações manualmente.

4. CONCLUSÕES

A aplicação otimizou consideravelmente o tempo gasto para se configurar uma rede nova e mantê-la documentada. Se tratando de redes já estabelecidas, há um tempo inicial gasto com a implantação do phpIPAM que é considerável, assumindo que centenas de dispositivos devem ser configurados. Ainda assim, há uma otimização ao considerar o longo prazo.

É necessário observar que, para que todas as etapas ocorram automaticamente, todos os servidores precisam ser corretamente configurados com o uso de chaves públicas e acesso SSH como usuário *root* liberado, já citados no artigo, assim como não deve ocorrer bloqueio da comunicação por *Firewalls*.

Diferente de outras soluções semelhantes, o software é capaz de enviar as configurações ao servidor DHCP e reiniciar o serviço *isc-dhcp* via SSH caso necessário, permitindo que o servidor seja configurado apenas com a utilização da aplicação, sem intervenção manual do administrador.

Para trabalhos futuros, destacam-se duas alternativas distintas: continuar a integração de outras funções do phpIPAM, como registros de DNS e integração com VLANs ou o desenvolvimento de uma ferramenta IPAM própria a partir da aplicação, com capacidade de configurar funções não previstas no phpIPAM, como *lease-time* definido por subrede, por exemplo.

Referências

ARAUJO, Gorgonio. **DHCP: Por que Usar?** 1997. Disponível em: <<https://memoria.rnp.br/newsgen/9705/n1-2.html>>. Acesso em: 10 jul. 2017.

INTERNET SYSTEMS CONSORTIUM. **ISC DHCP: Enterprise Grade Solution for Configuration Needs.** 2017. Disponível em: <<https://www.isc.org/downloads/dhcp/>>. Acesso em: 27 jul. 2017.

PHPIPAM. **PhpIPAM**: Open source IP address management. 2017. Disponível em: <<https://phpipam.net>>. Acesso em: 2 maio 2017.

SUSE. **Linux Continues to Gain Momentum in Enterprise Server Market.** 2013. Disponível em: <https://www.suse.com/docrep/documents/h7cvzy1buf/linux_gains_momentum.pdf>. Acesso em: 26 jul. 2017.

W3TECHS (Comp.). **Usage Statistics and Market Share of Server-side Programming Languages for Websites.** 2017a. Disponível em: <https://w3techs.com/technologies/overview/programming_language/all>. Acesso em: 01 ago. 2017.

W3TECHS (Comp.). **Usage Statistics and Market Share of Web Servers for Websites.** 2017b. Disponível em: <https://w3techs.com/technologies/overview/web_server/all>. Acesso em: 02 ago. 2017.

Apêndice A

Arquivo classes.php

```
<?php

require_once 'functions.php';
require_once 'config.php';

class Subnet{
    function __construct($description,$ipaddress,$mask,$sons,$id) {
        $this->id = $id;
        $this->description = $description;
        $this->ipaddress = $ipaddress;
        $this->mask = $mask;
        $this->sons = $sons;
    }

    function get_id(){
        return $this->id;
    }

    function get_description(){
        return $this->description;
    }

    function get_ipaddress(){
        return long2ip($this->ipaddress);
    }

    function get_mask(){
        return $this->mask;
    }

    function get_binary_mask(){
        return long2ip(pow(2, 32)-pow(2, 32-$this->mask));
    }

    function get_sons(){
        return $this->sons;
    }
}

class Host{
    function __construct($description,$ipaddress,$mac,$id) {
        $this->description = $description;
        $this->ipaddress = $ipaddress;
        $this->mac = $mac;
        $this->id = $id;
    }

    function get_description(){
        return $this->description;
    }
}
```

```

function get_ipaddress(){
return long2ip($this->ipaddress);
}

function get_mac(){
return $this->mac;
}
}
?>

```

Arquivo config.php

```

<?php
function conexao() {
$db_host = "192.168.1.28";
$db_name = "phpipam";
$db_user = "root";
$db_pass = "0oYiWPDNKBpFKSDS";

$conexao = new mysqli($db_host, $db_user, $db_pass, $db_name);
return $conexao;
}

function ssh_conect(){
return ssh2_connect("10.10.0.244");
}
?>

```

Arquivo functions.php

```

<?php
require_once "./classes.php";
require_once "./config.php";

function getSubnets($father = 0) {
$query = "SELECT * FROM subnets WHERE masterSubnetId = $father";
$result = conexao()->query($query);
$array_subnets = array();
while ($subnet = $result->fetch_object()) {
    $network = new Subnet(
        $subnet->description,
        $subnet->subnet,
        $subnet->mask,
        getSubnets($subnet->id),
        $subnet->id
    );
    $array_subnets[] = $network;
}
return $array_subnets;
}

function get_subnet($id){
$query = "SELECT * FROM subnets WHERE id = $id";
$result = conexao()->query($query);
while ($subnet = $result->fetch_object()) {

```

```

        $network = new Subnet($subnet->description,$subnet-
>subnet,$subnet->mask,getSubnets($subnet->id),$subnet->id);
        $network->nameservers = get_nameserver($subnet->nameserverId);
    }
    return $network;
}

function get_nameserver($id){
    $query = "SELECT * FROM nameservers WHERE id = $id";
    $result = conexao()->query($query);
    $nameservers = $result->fetch_object();
    return $nameservers;
}

function printSubnets($subnets){

    foreach ($subnets as $subnet) {
        echo "<div style='padding: 2px 0px 5px 15px; min-width:
300px;max-width: 400px'' >".
            "<div style='float: left'>".$subnet->get_ipaddress()." -
".$subnet->get_binary_mask()."</div>";
            if (count($subnet->sons) > 0){
                echo "<br>";
                echo "<div>";
                printSubnets($subnet->sons);
                echo "</div>";
            } else {
                echo "<div style='float: right; padding-right:10px'><input
type=\"checkbox\" name=\"subnets[]\" value=\"".$subnet-
>get_id()."\"></div><br>";
            }
            echo "</div>";

        }
    }

function getHosts($subnet) {
    $query = "SELECT * FROM ipaddresses WHERE subnetId = $subnet";
    $result = conexao()->query($query);
    $hosts = array();
    while ($host = $result->fetch_object()) {
        $obj_host = new Host($host->hostname, $host->ip_addr,$host-
>mac,$host->id);
        $hosts[] = $obj_host;
    }
    return $hosts;
}

function getHostsByState($subnet,$state) {
    $query = "SELECT * FROM ipaddresses WHERE subnetId = $subnet AND
state = $state";
    $result = conexao()->query($query);
    $hosts = array();
    while ($host = $result->fetch_object()) {
        $obj_host = new Host($host->hostname,$host->ip_addr,$host-
>mac,$host->id);
        $hosts[] = $obj_host;
    }
}

```

```

    return $hosts;
}

function getGateway($subnet){
    $query = "SELECT * FROM ipaddresses WHERE subnetId = $subnet AND
is_gateway = 1";
    $result = conexao()->query($query);
    $hosts = array();
    while ($host = $result->fetch_object()) {
        $obj_host = new Host($host->hostname,$host->ip_addr,$host-
>mac,$host->id);
        $hosts = $obj_host;
    }
    return $hosts;
}

function printDHCPConfig($subnet){
    return

    ""
    "subnet ".$subnet->get_ipaddress()." netmask ".$subnet-
>get_binary_mask()." {\n\n".
    print_gateway($subnet).
    print_dns_dhcp($subnet).
    print_range_dhcp($subnet).
    print_used_dhcp($subnet).
    "}".
    "\n\n";
}

function print_gateway($subnet){
    $gateway = getGateway($subnet->id);
    return "\toption routers ".$gateway->get_ipaddress().";\n";
}

function print_dns_dhcp($subnet){
    if(isset($subnet->nameservers)){
        $nameservers = str_replace(";", " ", $subnet->nameservers-
>namesrv1);
        return "\toption domain-name-servers ".$nameservers."; \n\n";
    }
    return;
}

function print_range($host){
    return "\trange ".$host->get_ipaddress()." ".$host-
>get_ipaddress().";\n";
}

function print_range_dhcp($subnet){
    $range_addresses = getHostsByState($subnet->get_id(),4);
    $str_range = "";
    foreach ($range_addresses as $host) {
        $str_range.=print_range($host);
    }
    return $str_range."\n";
}

```



```

$handle = fopen("./dhcpd.conf","w+");

dump_var($_POST);

if(isset($_POST["header"])){

$text=
"authoritative;".
"\ndefault-lease-time 600;".
"\nmax-lease-time 7200".
"\nsubnet 172.16.11.0 netmask 255.255.255.0 {".
"\n}\n\n";
fwrite($handle, $text);
}

foreach ($subnets as $subnet) {

$network = get_subnet($subnet);

// echo printDHCPConfig($network);

$text = printDHCPConfig($network);

fwrite($handle, $text);
}

fclose($handle);
header("location: index.php");

```

?>

Arquivo index.php

```

<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project
Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<?php
require_once 'classes.php';
require_once 'functions.php';
require_once 'config.php';
?>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body style="border: 0; margin: 0; padding: 0; font-family:
verdana; background-color: #CCC">
    <div style="margin: 0 auto; width: 900px;">
      <div style="background-color: rgb(40, 43, 46); color:
white; padding: 1px 20px;text-shadow: black 2px 2px 2px;">
        <h1>Exportação de Redes</h1>

```

```

        </div>
        <div style="background-color: #FFF">
            <div style="font-size: 10pt; padding: 10px 5px;
background-color: #FFF; float: left; width: 400px; ">
                <div style="text-align: center">
                    <h3 style="margin: 5px 0; ">Redes do phpIPAM</h3>
                    <hr>

                </div>
                <form action="build_dhcpd_conf.php" method="post">
                    <div style='padding: 2px 0px 5px 15px; min-width: 300px;max-
width: 400px'' >
                        <div style='float: left'>Cabecalho Inicial DHCP</div>
                        <div style='float: right; padding-right:10px'><input
type="checkbox" name="header" value=""></div><br>
                    </div>

                        <?php
                            $subnets = getSubnets();
                            printSubnets($subnets);
                            // printDHCPConfig($subnets[0]);
                            ?>
                        <div style="padding: 5px 20px">
                            <input type="submit" value="Gerar">

                        </div>
                    </form>
                </div>
                <div style="font-size: 10pt; padding: 10px 5px; float:
right; width: 480px; ">
                    <div style="text-align: center">
                        <h3 style="margin: 5px 0; ">Último DHCP
Gerado</h3>

                    <hr>

                    </div>
                    <div>
                        <?php
                            dump_DHCP_config();
                            ?>
                        <form action="send_dhcp_conf.php" method="post">
                            <div style="padding:
5px 20px">
                                <input type="submit" value="Enviar">

                            </div>
                        </form>
                    </div>
                </div>
                <div style="clear: both"></div>
            </div>
        </body>
    </html>

```