Red de Administración Remota

Camchong Kenming, Carrillo Christian, Salavarria Mayken, Tomalá Carlos
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
kcamchon@espol.edu.ec
ccarril@espol.edu.ec
mayusala@espol.edu.ec
cbtomala@espol.edu.ec

***Abstract*— Aplicación basada en el uso de un servidor bajo el sistema operativo de CentOs, el cual es un software de código abierto al alcance de todos. PHP lenguaje de programación usado en el servidor para poder solicitar, mediante una sesión mediante Telnet, información de los conmutadores y enrutadores que conforman la red como lo son la configuración básica, estado y tablas de enrutamiento, además de eso poder administrar los dispositivos cambiándoles la IP a las interfaces creación de Interfaces Loopback entre otras cosas.**

I. INTRODUCCIÓN

En la actualidad un Ingeniero de Networking debe estar al tanto del estado de los Enrutadores y Conmutadores en tiempo real, ya que si se presenta un problema en la red este debe hallar una solución óptima antes de que este pueda arribar al lugar donde se encuentra el dispositivo, cada minuto ocasiona pérdidas para la empresa proveedora de servicios.

II. CREACCION DE LA APLICACION

Se utiliza el Software Android Studio para darte la interfaz gráfica que la aplicación. Una de las aplicaciones más usadas para la creación de aplicaciones para Android. Fig.1

Fig.1 Android Studio

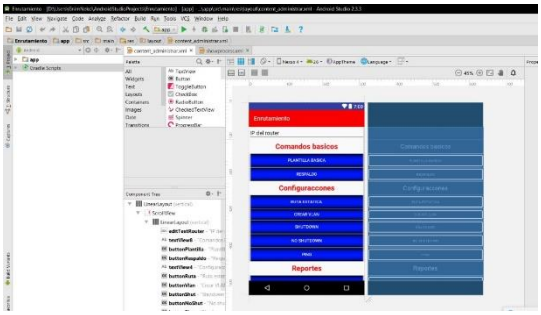

Fig.2 Interface Gráfica de la aplicación

III. SERVIDOR

Para esta parte del desarrollo del programa debemos crear un servidor que cumpla con los servicios que se desean desarrollar en la aplicación.
Esta va a ser desarrollada en CentOS

Fig.3 CentOS

IV. CODIGO PHP

El código PHP son scripts los cuales se encuentran dentro del servidor, ellos nos permitirán acceder a la información propia del enrutador o conmutador con el que queramos trabajar.

A. Configuración básica
B. Acceder a la tabla de enrutamiento
C. Creación de Loopbacks

Siendo estas algunas de las opciones de nuestra aplicación, a continuación, se detalla el código que se encuentra en el servidor.
Esta función guarda los logs de las personas que ingresan a la base de datos, los datos que se guardan son:

- Nombre
- IP
- Fecha y hora

```
require("config.inc.php");
function GetUserIP() {
            if (isset($_SERVER["HTTP_CLIENT_IP"]))
      {
        return $_SERVER["HTTP_CLIENT_IP"];
      }
      elseif
(isset($_SERVER["HTTP_X_FORWARDED_FOR"]))
      {
        return $_SERVER["HTTP_X_FORWARDED_FOR"];
      }
      elseif (isset($_SERVER["HTTP_X_FORWARDED"]))
      {
        return $_SERVER["HTTP_X_FORWARDED"];
      }
      elseif
(isset($_SERVER["HTTP_FORWARDED_FOR"]))
      {
        return $_SERVER["HTTP_FORWARDED_FOR"];
      }
      elseif (isset($_SERVER["HTTP_FORWARDED"]))
      {
        return $_SERVER["HTTP_FORWARDED"];
      }
      else
      {
        return $_SERVER["REMOTE_ADDR"];
      }
}
```

Esta función recibe como parámetros la ip del router al que vamos a configurar mediante telnet, el numero de la lookback que se va a crear, la ip y la máscara de la lookback que se va a crear.
Mediante la función fsockopen se ingresa al router a través telnet y se le mandan los comandos respectivos para la creación de una lookback con su ip y mascara.

```
function crearLoopBack($ip_router,$numero_loopback,$ip_loopback,$mascara_loopback){
$port = 23;
$timeout = 10;
$router_ip = $ip_router;
$username = "adm";
$password = "adm";
$connection = fsockopen($router_ip, $port, $errno, $errstr, $timeout);
if(!$connection){
return $ip_router;
} else {
 echo "Connected\n";
        fputs($connection, $username."\r\n");
        fputs($connection, $password."\r\n");
        fputs($connection, "config t\r\n");
        fputs($connection, "interface loopback".$numero_loopback."\r\n");
        fputs($connection, "ip address ".$ip_loopback." ".$mascara_loopback."\n");
        fputs($connection, "exit\r\n");
        $j = 0;
        while ($j < 16) {
                fgets($connection, 20);
                $j++;
        }
        stream_set_timeout($connection, 2);
        $timeoutCount = 0;
        return "Connected\n";
}
}
```

Esta función recibe como parámetros la ip del router al que vamos a configurar mediante telnet, la ip del router al cual queremos hacer ping y la ip origen para hacer un ping extendido.
Mediante la función fsockopen se ingresa al router a través telnet y se le mandan los comandos respectivos para realizar un ping extendido a la ip previamente ingresada.

```
function hacerPing($ip_router,$ip_destino,$ip_origen){
        $resultado="";
        $port = 23;
        $timeout = 10;
        $username = "adm";
        $password = "adm";
        $connection = fsockopen($ip_router, $port, $errno, $errstr, $timeout);
        if(!$connection){
                return "Error!";
        } else {
                fputs($connection, "$username\r\n");
                fputs($connection, "$password\r\n");
                fputs($connection, "ping ".$ip_destino." ".”source ”.$ip_origen."\r\n");
                fputs($connection, " ");
                $j = 0;
                while ($j < 7) {
                        fgets($connection, 128);
                        $j++;
                }
                stream_set_timeout($connection, 2);
                $timeoutCount = 0;

                $j=0;
                while ($j < 4) {
                        $resul=fgets($connection, 128);

$resultado=$resultado.$resul."\n";
                        $j++;
                }
                return $resultado;
        }
}
```

Esta función recibe como parámetros la ip del router al que vamos a configurar mediante telnet, la interface la cual queremos levantar.
Mediante la función fsockopen se ingresa al router a través telnet y se le mandan los comandos respectivos para levantar una interface mediante el comando no shutdown.

```
function shutdown($ip_router,$interface){
        $port = 23;
        $timeout = 10;
        $username = "adm";
        $password = "adm";
        $connection = fsockopen($ip_router, $port, $errno, $errstr, $timeout);
        if(!$connection){
                return $ip_router;
                exit();
        } else {
                echo "Connected\n";
               fputs($connection, $username."\r\n");
                fputs($connection, $password."\r\n");
                fputs($connection, "config t\r\n");
                fputs($connection, "interface ".$interface."\r\n");
                fputs($connection, "shutdown \r\n ");
                fputs($connection, "exit\r\n");
                $j = 0;
                 while ($j < 16) {
                        fgets($connection, 20);
                        $j++;
                 }
                 stream_set_timeout($connection, 2);
                 $timeoutCount = 0;
                 return "Connected\n";
        }
}
```

Este código ejecuta a las funciones previamente creadas dependiendo del parámetro "función" y varios "if", el cual nos dice a que función llamar. Aquí se reciben los parámetros que serán enviados a las funciones mediante mensajes JSON a través del comando $POST.

```
$ip = GetUserIP();


if (!empty($_POST)) {
        if ($_POST['funcion']==1){
                //obtenemos los usuarios respecto a la usuario que llega por parametro
                        $query = "
                                SELECT
                                   name,
                                   pass
                                FROM users
                                WHERE
                                   name = :username
                                      ";
                         $query_params = array(
                                        ':username' => $_POST['username']
                                        );


                try {
                        $stmt   = $db->prepare($query);
                        $result = $stmt->execute($query_params);
                }
                catch (PDOException $ex) {
                        //para testear pueden utilizar lo de abajo
                         //die("la consulta murio " . $ex->getMessage());

                 $response["success"] = 0;
                 $response["message"] = "Problema con la base de datos, vuelve a intetarlo";
                 die(json_encode($response));

                }

                 //la variable a continuación nos permitirará determinar
                 //si es o no la información correcta
                 //la inicializamos en "false"
                 $validated_info = false;

                 //vamos a buscar a todas las filas
                 $row = $stmt->fetch();
                 if ($row)
                 {
                        //encaso que no lo este, solo comparamos como acontinuación
                        if (md5($_POST['password']) == $row['pass'])
                                {
                                  $login_ok = true;
                            }
                }
 if ($login_ok)
                        {

                        $query = "
                                INSERT INTO
                                        log
                                        (name,IP,logTime)
                                VALUES
                                        (:username,:ip, NOW())
                                ";
                        $query_params = array(
                                ':username' => $_POST['username'],
                                ':ip'=>$ip
                                );
                         try
                        {
                                $stmt   = $db->prepare($query);
                                $result = $stmt->execute($query_params);
                        }
                          catch (PDOException $ex)
                        {
                        $response["success"] = 0;
                        $response["message"] = "Problema con la base de datos, vuelve a intentarlo";
                        die(json_encode($response));
                        }
                        $response["success"] = 1;
                        $response["message"] = "Login correcto!";
                        die(json_encode($response));
                }
                else
                {
                   $response["success"] = 0;
                   $response["message"] = "Login INCORRECTO";
                   die(json_encode($response));
                }
        }
```

```
elseif ($_POST['funcion']==2){ //loopback
                        $ip_router=$_POST['router'];
                        $numero_loopback=$_POST['numeroLoopback'];
                        $ip_loopback=$_POST['ip'];
                        $mascara_loopback=$_POST['mascara'];
                        $response["success"] = 1;
                        $response["message"]                                                                    =
crearLoopBack($ip_router,$numero_loopback,$ip_loopback,$mascara_loopback);
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==3){//ping
                        $ip_router= $_POST['router'];
                        $ip_destino=$_POST['ipDestino'];
                        $ip_origen=$_POST['ipOrigen'];
                        $response["success"] = 1;
                        $response["message"] ="Exito";
                        $resul = hacerPing($ip_router,$ip_destino,$ip_origen);
                 $response["text"]=$resul;
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==4){ //no shutdown
                        $ip_router=$_POST['router'];
                        $interface=$_POST['interface'];
                        $response["success"] = 1;
                        $response["message"] = noShutdown($ip_router, $interface);
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==5){ // shutdown
                        $ip_router=$_POST['router'];
                        $interface=$_POST['interface'];
                        $response["success"] = 1;
                        $response["message"] = shutdown($ip_router, $interface);
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==6){ //plantilla basica
                        $ip_router=$_POST['router'];
                        $response["success"] = 1;
                        $response["message"] = crearPlantillaBasica($ip_router);
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==8){ //crear ruta estatica
                        $ip_router=$_POST['router'];
                        $ip_destino=$_POST['ipDestino'];
                elseif ($_POST['funcion']==9){
                        $ip_router=$_POST['router'];
                        $response["success"] = 1;
                        $response["message"] ="Exito";
                        $resul = showRun($ip_router);
                 $response["text"]=$resul;
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==10){
                        $ip_router=$_POST['router'];
                        $response["success"] = 1;
                        $resul = showInterfaces($ip_router);
                 $response["text"]=$resul;
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==11){
                        $ip_router=$_POST['router'];
                        $response["success"] = 1;
                        $resul = showProcess($ip_router);
                 $response["text"]=$resul;
                        die(json_encode($response));
                }

                elseif ($_POST['funcion']==12){
                        $ip_router=$_POST['router'];
                        $response["success"] = 1;
                        $resul = showRoute($ip_router);
                 $response["text"]=$resul;
                        die(json_encode($response));
                }
}
else
{
        ?>
         <h1>Login</h1>
         <form action="index.php" method="post">
            Username:<br />
            <input type="text" name="username" placeholder="username" />
            <br /><br />
            Password:<br />
            <input type="password" name="password" placeholder="password" value="" />
```

```
<br /><br />
            <input type="submit" value="Login" />
          </form>
          <a href="register.php">Register</a>
          <?php
}
```

Fig. 4 Configuración de la aplicación

V.PRUEBAS

En este tipo de conexiones siempre se debe tomar mucho en cuenta tener los permisos necesarios para poder ingresar sin ningún contratiempo a los dispositivos ya que mucho de estos poseen claves de acceso. Además de esto debemos bajar el fireware de la PC de la cual estamos usando de servidor para que podamos tener conexión exitosa.

En este caso si no poseemos el dispositivo podemos usar una herramienta auxiliar para probar si la aplicación es un éxito, esta es GNS3 la cual es un simulador de redes con una interfaz muy similar a los dispositivos reales. Fig. 4

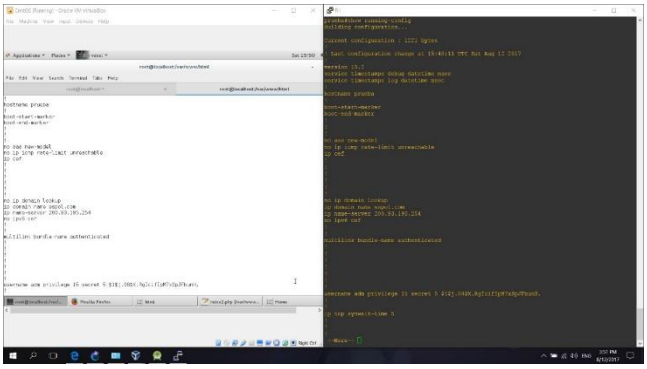

Fig. 5 Pruebas.

VII.CONCLUSIONES

La creación de aplicaciones de monitoreo y configuración, son muy demandadas y comúnmente desarrolladas en la actualidad, puesto que estas pueden ser desarrolladas utilizando softwares libres que ocasionan bajos costos, siendo estas de gran utilidad para un mejor desempeño en el campo laboral de los Ingenieros de Networking.

REFERENCIAS

[1] http://php.net/manual/es/intro-whatis.php
[2] http://php.net/manual/es/intro-whatcando.php
[3] https://www.gns3.com/software