

Creación de un Portal Cautivo



Por Helena Palos Alonso

Contenido

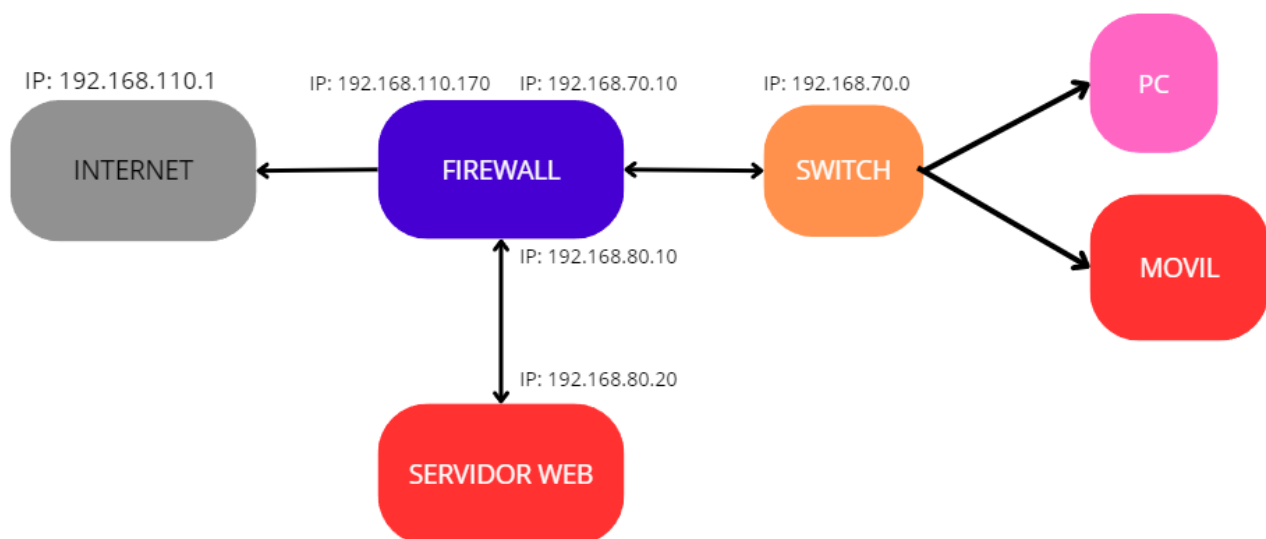
Contenido	2
1. Objetivo	3
2. Motivacion.....	4
3 Documentacion	5
3.1 Creacion de las primeras normas	5
3.2 Creacion del servidor DHCP	8
3.3 Creacion de la pagina web.....	10
3.4 Script de Python	15
3.5 Version de pago	17
3.6 Usuarios Windows.....	20
3.6 Usuarios Android.....	21
3.8 Pagina para listar a los usuarios y kickearlos.....	24
3.10 Tickets.....	33
3.12 SSL a la pagina web.....	41
3.13 Todo unido.....	43
3.14 Usuarios Bloqueados (Extra)	47
3.15 Servidor DNS + Proxy	52
FALLOS Y COMO RESOLVERLOS.....	58
4.Aplicabilidad	60
4.1 Propuesta de valor	60
4.2 .Objetivo	60
4.3. Usuario y Cliente final	61
4.4 Usabilidad	62
4.5 Intermediario / Comprador / Promotor	62
4.6 Puntos fuertes y débiles	62
4.7 Prospectiva	63
4.8 Presupuesto.....	63
4.9 Previsio temporitzacio de tasques	65
4.9 Conclusion	66
5.Bibliografia.....	67
6.Anexos	68

1. Objetivo

El objetivo de este proyecto es el de crear un portal cautivo, y se preguntará, que es esto ? Es un servicio provisto para dar internet en sitios públicos el cual al recibir la señal esta nos redireccionará a una web donde tendremos que hacer algunas acciones como un registro previo, puede tener opción a pago pero normalmente es gratis, la opción de pago tendrá mejoras como mas ancho de banda o mas tiempo de uso, ahora bien el crear esto desde cero y hacerlo funcional va a ser un gran camino que voy a recorrer. Este es el objetivo básico luego se harán mejoras como soporte a Android , el capar la velocidad entre otras cosas.

El portal cautivo esta compuesto de 2 servidores uno que actuará de firewall el y el otro que actuará como servidor, web, el proyecto lo realizaré con Iptables , PHP y Python entre otras cosas combinándolas para así crear el sistema.

Los usuarios se conectaran al firewall mediante un switch el cual repartirá a cada cliente una ip cuando se conecta, gracias al servidor DHCP que tiene el firewall.



2. Motivacion

Mi motivación fue que durante años he estado viendo estos sistemas que te redireccionan solo con dar un botón y registrándose desde que era pequeña siempre me he preguntado. ¿cómo funcionaban? ¿Qué hacen? ¿Como hace esto? Hasta que hace poco descubrí como podríamos hacer esto, ahí es donde cogí esta oportunidad, como funciona la redirección, como se pueden usar para dirigir el marketing, como dar seguridad a una red publica, esa fue mi mayor motivació

3 Documentacion

3.1 Creacion de las primeras normas

Primeramente como en todos los servidores configuramos la red con netplan para luego aplicarla

```
# This is the network config written by 'subiq
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.110.170/24]
      gateway4: 192.168.110.1
      nameservers:
        addresses: [8.8.8.8]
    enp0s8:
      dhcp4: false
      addresses: [192.168.70.10/24]
      nameservers:
        addresses: [192.168.110.170]

version: 2
```

Luego desarrollamos la primera política la cual es de denegación total

```
$ ipt -F
$ ipt -X
$ ipt -Z
$ ipt -t nat -F
$ ipt -t nat -X
$ ipt -t nat -Z
$ ipt -P INPUT DROP
$ ipt -P OUTPUT DROP
$ ipt -P FORWARD DROP
$ ipt -t nat -P PREROUTING ACCEPT
$ ipt -t nat -P POSTROUTING ACCEPT

#comunicacion local
```

Luego se añade el que se acepten los paquetes http y https esto se hará para poder navegar por internet sin problema

3.Documentacion

```
$ipt -A FORWARD -i enp0s8 -s 192.168.70.0/24 -p tcp --dport 80 -j ACCEPT
$ipt -A FORWARD -i enp0s8 -s 192.168.70.0/24 -p tcp --dport 443 -j ACCEPT
$ipt -A FORWARD -i enp0s3 -p tcp --sport 80 -d 192.168.70.0/24 -j ACCEPT
$ipt -A FORWARD -i enp0s3 -p tcp --sport 443 -d 192.168.70.0/24 -j ACCEPT
```

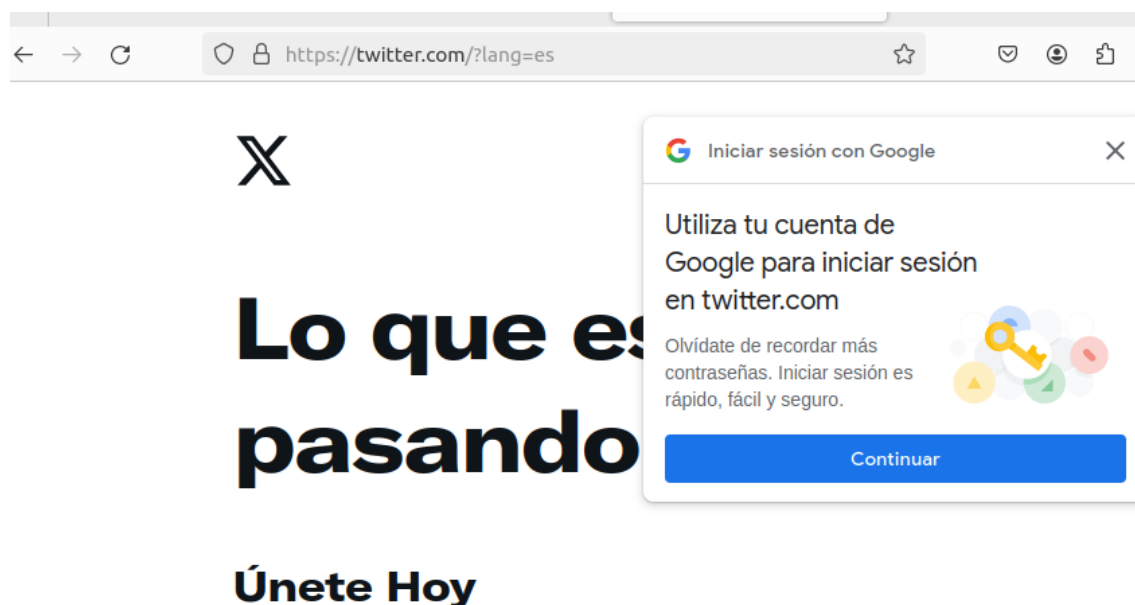
Se introducen las reglas sobre el DNS para que vayan al dns el cual es el 192.168.110.1

```
# Permitir trafico DNS
$ipt -A FORWARD -i enp0s8 -s 192.168.70.0/24 -p udp --dport 53 -d 192.168.110.1 -j ACCEPT
$ipt -A FORWARD -i enp0s3 -s 192.168.110.1 -p udp --sport 53 -d 192.168.70.0/24 -j ACCEPT
$ipt -A FORWARD -i enp0s8 -s 192.168.70.0/24 -p udp --dport 53 -d 8.8.4.4 -j ACCEPT
$ipt -A FORWARD -i enp0s3 -s 8.8.4.4 -p udp --sport 53 -d 192.168.70.0/24 -j ACCEPT
```

Y se activa el redireccionamiento

```
$ipt -t nat -A POSTROUTING -s 192.168.70.0 -o -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ahora podemos acceder sin problema a una pagina web



Ahora ponemos un permiso entre la dmz y la lan que crearemos posteriormente

```
$ipt -A FORWARD -i enp0s8 -p tcp -d 192.168.80.20 --dport 80 -j ACCEPT
$ipt -A FORWARD -i enp0s9 -p tcp -d 192.168.80.20 --sport 80 -j ACCEPT
$ipt -t nat -A PREROUTING -i enp0s8 -p tcp -d 192.168.70.10 --dport 80 -j DNAT --to 192.168.80.20:80
```

3.Documentacion

Para luego añadir el portal cautivo

```
$ iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 80 -j DNAT --to 192.168.80.20:80

#portal cautivo

$ iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 80 -j DNAT --to 192.168.80.20:80
```

Ahora una vez añadido cuando tengamos el portal nos aparecerá el pop up de que se necesitaría iniciar sesión, luego añadimos una norma que todos los paquetes sean marcados y los cuales no sean marcados como 1

```
# Crear la cadena "internet" en la tabla mangle
$ iptables -t mangle -A PREROUTING -i enp0s8 -s 192.168.70.10 -j MARK --set-mark 1
$ iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 80 -m mark ! --mark 1 -j DNAT --to 192.168.80.20:80
$ iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 443 -m mark ! --mark 1 -j DNAT --to 192.168.80.20:443
```

esto fue un desastre tardé varios días intentando hacerlo de tantas maneras que ya me estaba desesperando, ya que ninguna funcionaba como quería pero gracias a nuestro compañero Jordi se pudo solucionar y seguir adelante, así que gracias Jordi.

Después de solucionar este problemilla tenía las cosas más claras así que seguí adelante con el proyecto

3.Documentacion

3.2 Creacion del servidor DHCP

Ahora descargamos el servidor DHCP y lo configuramos, el sistema recae en que haya un router por lo cual no haremos servidor DNS

```
helena@helena:/etc/iptables$ sudo apt-get install isc-dhcp-server
```

Configuramos el fichero /etc/default/isc-dhcp-server

```
# On what interfaces should the
# Separate multiple inter
INTERFACESv4="enp0s8"
INTERFACESv6=""
```

Configuramos el fichero /etc/dhcp/dhcpd.conf con la red 192.168.70.0 con un rango de 100 ips y con el DNS el cual es el 192.168.110.1

```
# Configuración global
authoritative;
subnet 192.168.70.0 netmask 255.255.255.0 {
    range 192.168.70.100 192.168.70.200;
    option routers 192.168.70.10;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
default-lease-time 600;
max-lease-time 7200;
```

Esto hará que podamos crear el servidor con ip tables junto a un dns

Ahora bien nos quedan algunas cosillas que hacer aun. Podremos ver como nuestro cliente se conecta por dhcp

Cancelar

Perfil 5

Aplicar

Detalles

Identidad

IPV4

IPV6

Seguridad

Método IPv4

☒ Automático (DHCP)

☐ Sólo enlace local

☐ Manual

☐ Desactivar

☐ Compartida con otros equipos

DNS

Automático ☒

Direcciones IP separadas por comas

Cancelar

Perfil 5

Aplicar

Detalles

Identidad

IPV4

IPV6

Seguridad

Velocidad de conexión

1000 Mb/s

Dirección IPv4

192.168.70.186

Dirección IPv6

fe80::8095:ae7:9ef4:e7a6

Dirección física

08:00:27:CA:25:C9

Ruta predeterminada

192.168.70.10

DNS

8.8.8.8 8.8.4.4

3.Documentacion

Ahora vamos de nuevo a las interfaces y añadimos otra la cual es nuestro servidor web

```
addresses: [192.168.110.170]
enp0s9:
  dhcp4: false
  addresses: [192.168.80.10/24]
  nameservers:
    addresses: [192.168.110.170]
```

Y vemos que si lo aplicamos lo vemos

```
valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
up default qlen 1000
    link/ether 08:00:27:12:c1:37 brd ff:ff:ff:ff:ff:ff
    inet 192.168.110.170/24 brd 192.168.110.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe12:c137/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
up default qlen 1000
    link/ether 08:00:27:1c:cf:38 brd ff:ff:ff:ff:ff:ff
    inet 192.168.70.10/24 brd 192.168.70.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1c:cf38/64 scope link
        valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
up default qlen 1000
    link/ether 08:00:27:b2:6d:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.80.10/24 brd 192.168.80.255 scope global enp0s9
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb2:6db8/64 scope link
        valid_lft forever preferred_lft forever
helena@helena:/etc/iptables$
```

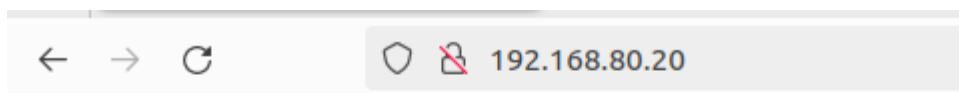
Pero primero tenemos que montar el servidor web el cual estará apartado del resto con una red 192.168.80.20 con la puerta de enlace esta siendo la 192.168.80.10

3.3 Creacion de la pagina web

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>

<head>
</head>
<body>
    <p>MUCHO TEXTO </p>
</body>
</html>
```

Ahora vamos al cliente y entramos, podemos visualizar que funciona la conexión entre el servidor web el router y el cliente y podemos ver que la norma funciona correctamente



MUCHO TEXTO



Ahora iremos a detallar el servidor web para ello tenemos que instalar PHP para que así realice las conexiones al servidor Mysql

3.Documentacion

```
/usr/bin/xauth: file /home/helena/.xauthrc does not exist
helena@helena:~$ sudo apt-get install php8.1
[sudo] password for helena:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.1 php-common php8.1-cli php8.1-common php8.1-opcache
  php8.1-readline
Paquetes sugeridos:
```

Y luego lo activamos

```
helena@helena:~$ sudo a2enmod php8.1
Considering dependency mpm_prefork for php8.1:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php8.1:
Module php8.1 already enabled
helena@helena:~$
```

Cambiamos el archivo por un php

```
helena@helena:~$ sudo mv /var/www/html/index.html /var/www/html/index.php
helena@helena:~$
```

Y ahora vamos a crear el archivo php por dentro junto a un CSS

Ahora vamos a instalar el mysql en el servidor web

```
helena@helena:~$ sudo apt-get install mysql-server
[sudo] password for helena:
```

Y luego en el server web instalamos el mysql y habilitamos el acceso de externos a este

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '123';
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY '123';
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)
```

Ahora bien creamos la base de datos

3.Documentacion

```
mysql> create database Base;
Query OK, 1 row affected (0,02 sec)

mysql>
```

Ahora vamos a los ajustes y comentamos esta parte de /etc/mysql/mysql.conf.d/mysqld.cnf

```
# Instead of skip-networking the default is now to listen on
# localhost which is more compatible and is not less secure
#bind-address            = 127.0.0.1
#mysqlx-bind-address     = 127.0.0.1
#
```

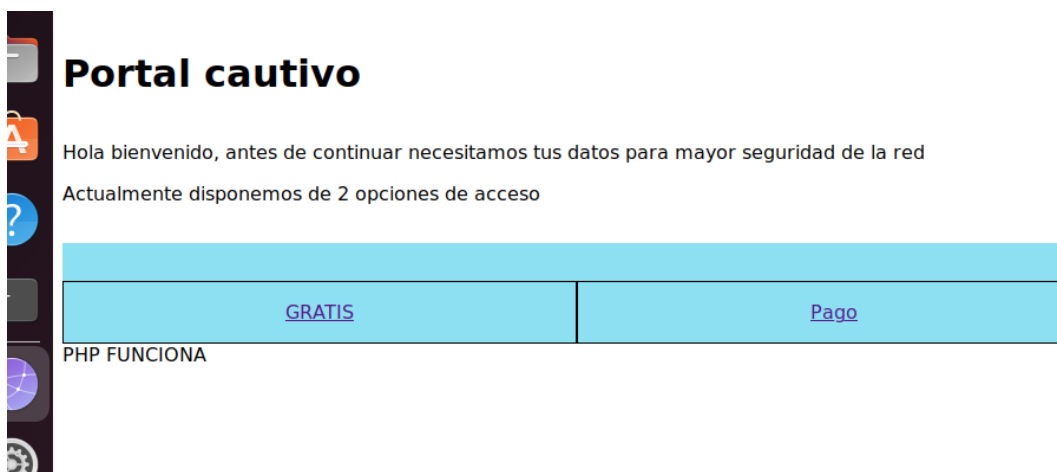
Y lo reiniciamos

```
helenahelena:~$ sudo systemctl restart mysql.service _
```

Una vez aplicado añadimos en el PHP que busque en la base de datos y esto una vez aplicado podemos ver que...

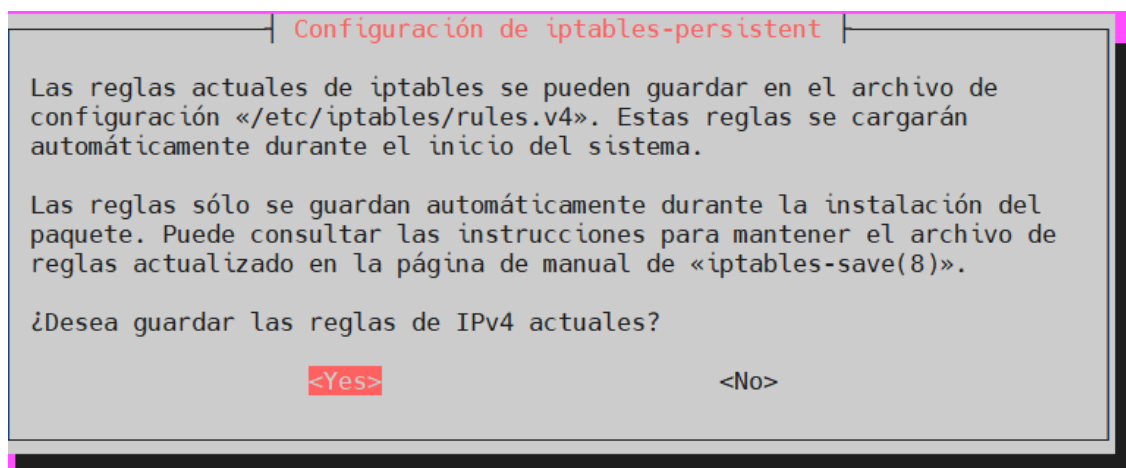
```
<?php
    echo "PHP FUNCIONA";
?>
```

Puede acceder sin problema ya que se carga la web esto no sucedería si no funcionara

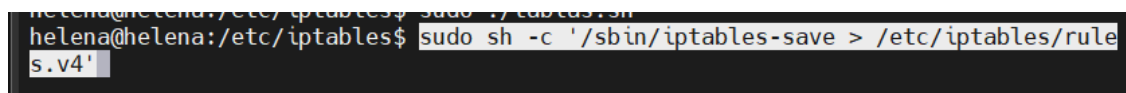


Ahora instalamos el ip tables persistant al tener reglas aplicamos esto

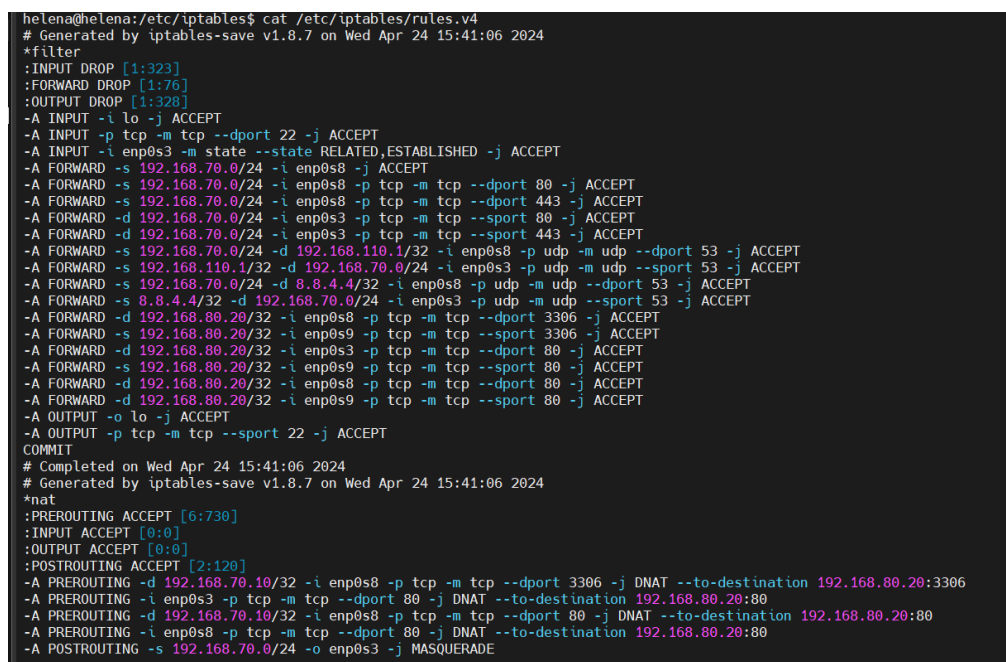
3.Documentacion



Ahora aplicamos las reglas buenas



Y podemos ver que si las vemos funcionan



Ahora volvamos a la base de datos y creamos el registro en la base de datos la cual tendrá el nombre “Registro de Conexiones”

3.Documentacion

```
mysql>
mysql> CREATE TABLE registro_conexiones (
->   id INT AUTO INCREMENT PRIMARY KEY,
->   nombre VARCHAR(100) NOT NULL,
->   correo VARCHAR(100) NOT NULL,
->   ip VARCHAR(45) NOT NULL,
->   mac VARCHAR(17) ,
->   duracion TIME,
->   tipo VARCHAR(20) NOT NULL,
->   fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
-> );
Query OK, 0 rows affected (0,06 sec)

mysql> █
```

Ahora nos adentramos y nos ponemos a hacer la pagina

Ahora añadimos que se haga un registro en la parte Gratis para que se añada a la base de datos de que se ha añadido este usuario con esta ip

```
$nom = $_POST['nombre'];
$correo = $_POST['email'];

$insertar = "INSERT INTO registro_conexiones (nombre, correo, ip, duracion, tipo, fecha)
VALUES ('$nom', '$correo', '$ip', NULL, 'gratis', '$hora')";

$resultado = mysqli_query($enlace, $insertar);

if (!$resultado) {
    echo "Registro fallido ha habido un fallo en el sistema no tte preocupes <br>";
} else {
    echo "Registro completado, espera unos momentos y puedes ya puedes abandonar la
pagina";
    echo $mac;
}
```

```
mysql> SELECT * FROM registro_conexiones;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | nombre | correo          | ip          | mac | duracion | tipo | fecha          |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | helena | heelna@wfwef | 192.168.70.188 | NULL | NULL | gratis | 2024-04-24 23:02:05 |
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```

3.4 Script de Python

Ahora falta lo complicado hacer un script para que lo acepte y que guarde esa mac

Esto por ahora solo obtiene la ip del dispositivo y luego hace un arp para obtener la mac y asi hacer una regla de aceptación para esta

```
import subprocess
import mysql.connector
import time

iptables = "/sbin/iptables"

# Función para ejecutar comandos iptables
def run_iptables_command(command):
    subprocess.run(command, shell=True, check=True)

while True:
    # Conexión a la base de datos
    conexion1 = mysql.connector.connect(host="192.168.80.20", user="root", passwd="123")
    cursor1 = conexion1.cursor()
    cursor1.execute("USE Acceso")
    cursor1.execute("SELECT * FROM registro_conexiones")

    resultados = cursor1.fetchall()

    if resultados: # Verificar si hay resultados
        for base in resultados:
            ip = base[3]
            total = f"arp -n {ip}"

            # Ejecutar comando arp
            aprobado = subprocess.run(total, shell=True, check=True, capture_output=True, text=True)
            salida = aprobado.stdout
            print(salida)
            lineas = salida.splitlines()
            linea = lineas[1]
            linea = linea.split()
            print(linea)
            mac = linea[2]
            print(mac)
            inserta="UPDATE registro_conexiones SET mac = %s WHERE ip = %s"

            datos=(mac,ip,)
            cursor1.execute(inserta, datos)
            # Cerrar la conexión actual
            conexion1.close()
```

Esto se hace aparte por que el mismo servidor web no puede manejar el arp ya que esta conectado a otra interfaz y no detecta a los usuarios conectados en el firewall

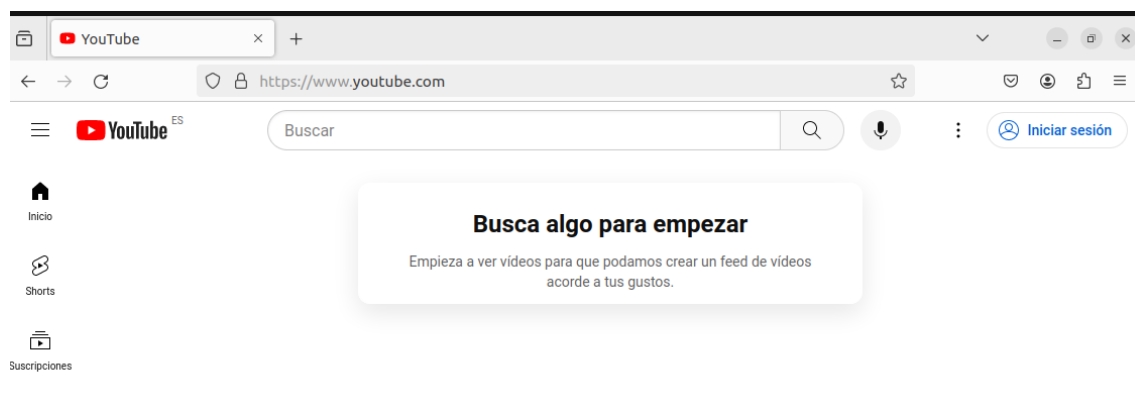
Ahora añadimos un comando para la tabla mangle

```
coma=f"sudo {iptables} -t mangle -A PREROUTING -m mac --mac-source {mac} -j MARK --set-mark 1"
run_iptables_command(coma)
print(coma)
```

El cual ejecuta el comando de todo lo que sea de cierta mac sea marcado como uno y eso significa que puede hacer su trayecto normal

3.Documentacion

ver una demostración del funcionamiento, ahora bien necesitamos ajustar algunas cosillas



Pero vamos en buena dirección

Este comando se usa para borrar todo

```
sudo iptables -t mangle -F
```

```
sudo iptables -t mangle -X
```

```
sudo iptables -t mangle -P INPUT ACCEPT
```

```
sudo iptables -t mangle -P OUTPUT ACCEPT
```

```
sudo iptables -t mangle -P FORWARD ACCEPT
```

Ahora vamos a la base de datos y añadimos otra tabla

```
Database changed
mysql> CREATE TABLE exusuarios (
->   id INT AUTO_INCREMENT PRIMARY KEY,
->   mac VARCHAR(17) NOT NULL,
->   nombre VARCHAR(100) NOT NULL,
->   tiempo_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
->   tiempo_expiracion TIMESTAMP NOT NULL,
->   INDEX(mac)
-> );
Query OK, 0 rows affected (0,30 sec)

mysql>
```

Este es para registrar los usuarios conectados , la cual introduciremos la query para que introduzca los datos de los usuarios que han entrado ya:

El cual nos dará todo:

3.Documentacion

```
index.html: capture index out of range
helena@helena:/etc/iptables$ python3 /home/helena/ejecucion.py
No hay datos en la tabla de registros.
(47, '08:00:27:f3:6a:17', 'qewqeqw@weqwew', datetime.datetime(2024, 4, 29, 16, 13
(48, '08:00:27:f3:6a:17', 'dfsdfas@ewrewr', datetime.datetime(2024, 4, 29, 16, 2
No hay datos en la tabla de registros.
(47, '08:00:27:f3:6a:17', 'qewqeqw@weqwew', datetime.datetime(2024, 4, 29, 16, 13
(48, '08:00:27:f3:6a:17', 'dfsdfas@ewrewr', datetime.datetime(2024, 4, 29, 16, 2
No hay datos en la tabla de registros.
(47, '08:00:27:f3:6a:17', 'qewqeqw@weqwew', datetime.datetime(2024, 4, 29, 16, 13
(48, '08:00:27:f3:6a:17', 'dfsdfas@ewrewr', datetime.datetime(2024, 4, 29, 16, 2
No hay datos en la tabla de registros.
```

Ahora tenemos que añadir lo que sucede cuando el tiempo pasa su limite

Debajo añadiremos un if que mira según el tiempo que ha pasado y si es mas pequeño que el tiempo actual, si es asi , se quitará sus permisos agarrando la mac que esta guardada en el sistema y marcando sus paquetes como 0 osea sin marcar.

```
print(base)
if fecha_actual > base[3]:
    mac=base[2]
    print("se acabo el tiempo")
    coma=f"sudo {iptables} -t mangle -A PREROUTING -m mac --mac-source {mac} -j MARK --set-mark 0"
    run_iptables_command([coma])
```

Y luego borraremos el usuario de exusuarios para

```
conexion2 = mysql.connector.connect(host="192.168.80.20", user="root", passwd="123")
cursor2 = conexion2.cursor()
cursor2.execute("USE Acceso")
borrar = "DELETE FROM exusuarios WHERE mac = %s"
cursor2.execute(borrar, (mac,))
conexion2.commit()
conexion2.close()
```

El invocar el aviso es diferente, se realizará mas adelante esto.

3.5 Version de pago

Vamos a inicio y añadimos la opción de pago dentro del php añadiendo el enlace de la pagina la cual será la de pago

```
<a class="color" href="/registro.php"><p>DATAS</p></a>
</div>
<div class="cubo2">
    <a class="color" href="/pago.php"><p>PAGO</p></a>
```

Luego creamos el archivo, el cual es el mismo que el anterior pero se le añaden varias cosas, como los campos para indicar que se tiene que realizar un pago, como el numero de la tarjeta, el cvv, la fecha de vencimiento entre otras cosas, esto hará que una vez se haga esto pueda continuar (obviamente no se realizará ningún pago esto es una simulación) e indicamos que todo es obligatorio

3.Documentacion

```
<h3>Datos de la Tarjeta</h3>
<label for="nombre_tarjeta">Nombre en la Tarjeta:</label><br>
<input type="text" id="nombre_tarjeta" name="nombre_tarjeta" required><br>

<label for="numero_tarjeta">Número de Tarjeta:</label><br>
<input type="text" id="numero_tarjeta" name="numero_tarjeta" maxlength="16" required><br>

<label for="fecha_vencimiento">Fecha de Vencimiento:</label><br>
<input type="text" id="fecha_vencimiento" name="fecha_vencimiento" placeholder="MM/YY" required><br>

<label for="cvv">CVV:</label><br>
<input type="text" id="cvv" name="cvv" maxlength="3" required><br><br>

<label type="hidden" id="pago"> </label>
<input type="submit" value="Pagar">
```

The screenshot shows a web form with a blue background. It is divided into two main sections: 'Datos del Usuario' and 'Datos de la Tarjeta'. The 'Datos del Usuario' section has two input fields: 'Nombre:' and 'Correo electrónico:'. The 'Datos de la Tarjeta' section has four input fields: 'Nombre en la Tarjeta:', 'Número de Tarjeta:', 'Fecha de Vencimiento:' (with a placeholder 'MM/YY'), and 'CVV:'. At the bottom of the form is a green button labeled 'Pagar'.

Ahora vamos a el cliente a comprobarlo. Y podremos ver que esto quedará así, ya que comparten css.

3.Documentacion

Añadimos a ambos php un atributo hidden para asi diferenciarlos

Este seria el de pago

```
<input type="hidden" id="tipo" name="tipo" value="pago"> </input>
```

Y este el gratis

```
<input type="hidden" id="tipo" name="tipo" value="gratis"> </input>
```

Ambos serán enviados en el php para asi identificar su tipo, ahora vamos al Python y creamos una separación de usuarios gratis y de pago , esto lo obtendremos de la base de datos el cual será el que registre todo, esto hará que los usuarios gratis, tendrán 1 hora de máximo y los usuarios de pago tendrán 5 horas.

Este es el código de los usuarios gratis.

```
if tipo=="gratis":  
    fecha_actual = datetime.now()  
    una_hora = timedelta(hours=1)
```

Y este el de los usuarios de pago.

```
elif tipo=="pago":  
    fecha_actual = datetime.now()  
    una_hora = timedelta(hours=5)
```

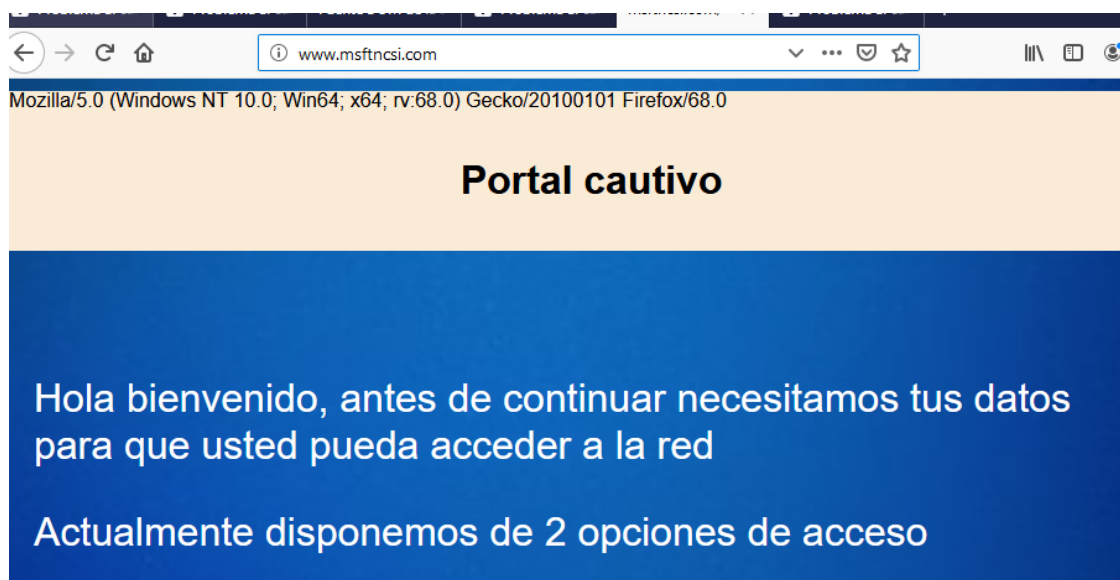
Veis que solo cambian las horas por lo tanto no hace falta modificar el código anterior, ahora entramos y vemos que tendremos un tiempo de 5 h

```
sudo /usr/sbin/iptables -t mangle -A PREROUTING -i mac  
2024-05-02 23:46:47.474845 2024-05-02 18:46:47.474845  
Error: Exclusivity flag on, cannot modify.
```

3.6 Usuarios Windows

Para que los usuarios Windows puedan acceder sin problemas normalmente usan una url : la cual actúa como redirección para estos portales cautivosañadiremos un rewrite en el archivo `/etc/apache2/sites-enabled/000-default.conf` para que redirija las solicitudes a esta url para que vaya hacia nuestra ip (no funciona, en ethernet)

```
RedirectMatch 302 /ncsi.txt http://192.168.70.10/index.php
```

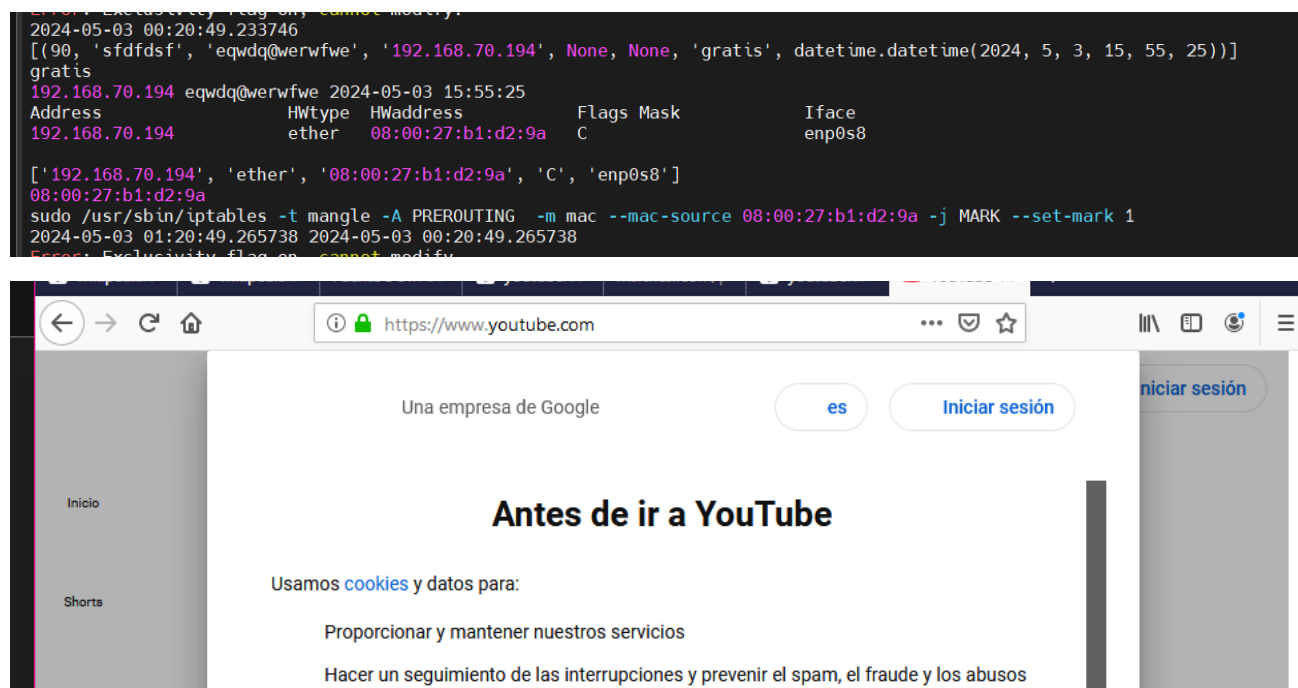


Realizamos el registro normal como cualquier usuario

A screenshot of a user registration form. The form has a light orange header with the text 'Registro de Usuario'. The main content area is blue and contains a white registration box. Inside the box, there are two input fields: 'Nombre:' with the value 'wdawdwdqweewq' and 'Correo electrónico:' with the value 'qweqweqwe@weqweq'. Below the input fields is a green button labeled 'Registrarse'.

3.Documentacion

Y podremos ver que hace el registro adecuadamente



Vemos que podemos acceder a youtube normalmente, por desgracia no es automatico pero podríamos aplicar una indicación que accedan a cierta url : <http://www.msftncsi.com> o a la ip 192.168.70.10 me temo que como no tenemos Firefox no funciona tan bien.

3.6 Usuarios Android

Para los usuarios Android añadiremos un rewrite en el archivo /etc/apache2/sites-enabled/000-default.conf el cual causará que cuando Android haga una solicitud para comprobar si tiene internet o no se encontrará de bruces con nuestro portal cautivo

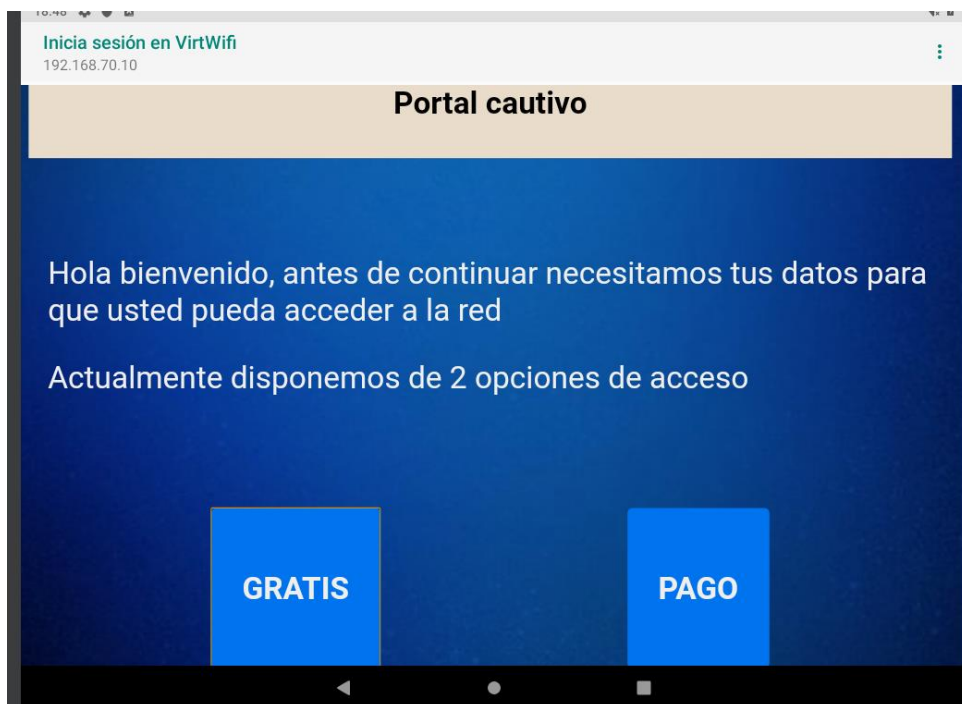
```
# android
    RedirectMatch 302 /generate_204 http://192.168.70.10/index.php
# android
```

Esto causará que podamos ver esto en el cliente Android, nos pide iniciar sesión

3.Documentacion



Una vez le clicamos entonces nos saldrá esto el cual es el portal cautivo sin problemas



Ahora si creamos una cuenta e iniciamos podemos ver que se ha creado el registro

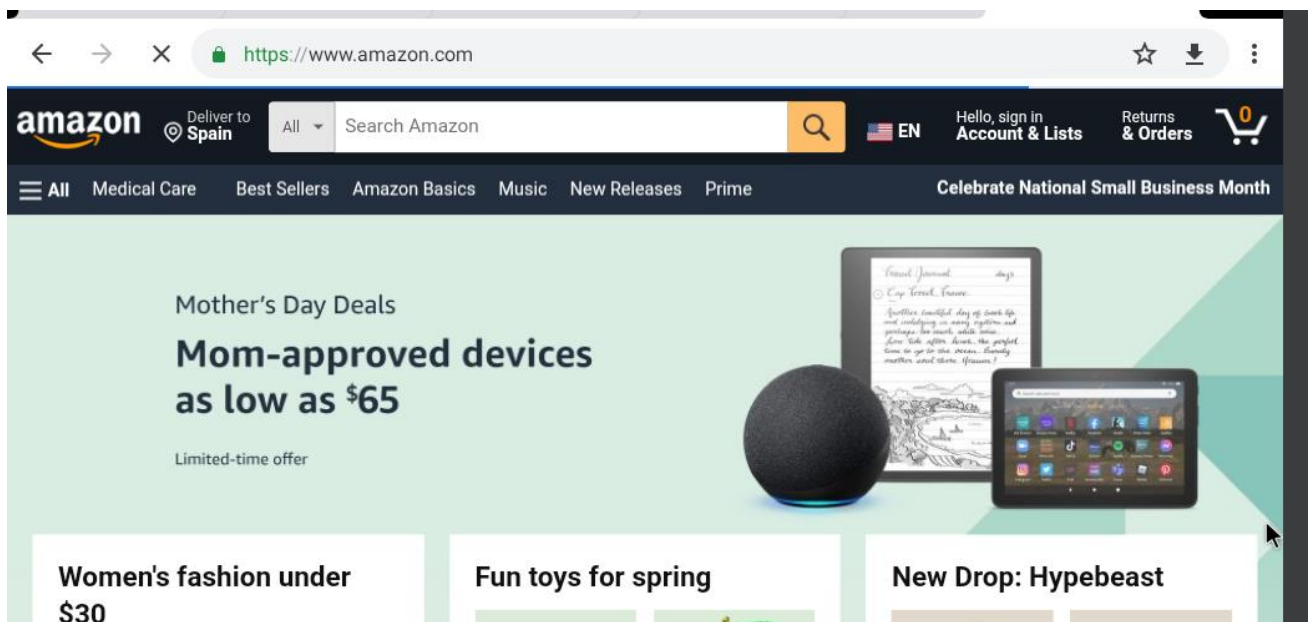
```
[ '192.168.70.196', 'ether', '08:00:27:cb:29:62', 'C', 'enp0s8' ]
08:00:27:cb:29:62
[sudo] password for helena:
sudo /usr/sbin/iptables -t mangle -A PREROUTING -m mac --mac-source 08:00:27:cb:29:62 -j MARK --set-mark 1
no hay registros
[(84, '08:00:27:b1:d2:9a', '192.168.70.194', 'gratis', '192.168.70.194', datetime.datetime(2024, 5, 6, 13, 37, 53), datetime.datetime(2024, 5, 6, 14, 37, 53)),
```

Y luego de unos instantes tendremos conexión



3.Documentacion

Y podremos ver que funciona correctamente



3.8 Pagina para listar a los usuarios y kickearlos

Primero tenemos que hacer una base de datos donde guardaremos los antiguos usuarios que se conectaron y acabó su tiempo.

```
CREATE TABLE IF NOT EXISTS antiguos_usuarios(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    ip VARCHAR(15) NOT NULL,  
    nombre VARCHAR(50) NOT NULL,  
    mac VARCHAR(17) NOT NULL,  
    fecha DATE NOT NULL  
);
```

Ahora que hemos acabado los accesos vamos a listar a los usuarios que hay, esto se hará con php y mysql ahora bien primero haremos un formulario para que tenga el administrador un registro de todo

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  
    <title>Registro de Usuario</title>  
    <link rel="stylesheet" href="Css/Registro.css">  
</head>  
<body>  
    <div class="cuadrado">  
        <h2>Acceso Administrador</h2>  
        </div>  
        <form action="process_registro.php" method="POST">  
            <label for="nombre">Nombre:</label>  
            <input type="text" id="nombre" name="nombre" required>  
  
            <label for="contra">Contraseña:</label>  
            <input type="password" id="contra" name="contra" required>  
  
            <input type="submit" value="Registrarse" >  
        </form>  
    </body>
```

Luego si le damos click llegaremos a este enlace el cual solo nos redireccionará o si falla volverá a la pagina anterior

3.Documentacion

```
php
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="Css/Principal.css">
</head>
<body>
<?php
    $nom = $_POST['nombre'];
    $contraseña = $_POST['contra'];
    if ($nom == 'pacos' and $contraseña=='123'){
        header("Location: pantalla.php ");
    }
    else{
        header("Location: lista.php ");
    }
}
```

entonces se pasará a la tabla donde mostrará a todos los usuarios

```
<?php

    session_start();
    // Establecer la conexión a la base de datos
    $enlace = mysqli_connect("192.168.80.20", "root", "123", "Acceso");
    if (!$enlace) {
        echo "<p class='error'>Error en la base de datos: " . mysqli_connect_error()
. "</p>";
        exit;
    }

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="Css/tabla.css">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,F
ILL,GRAD@20..48,100..700,0..1,-50..200" />
    <title>Usuarios Registrados</title>
</head>
<body>
    <div class="container">
        <h1>Usuarios Registrados</h1>
        <?php
```

```
$sql = "SELECT * FROM exusuarios";
$datos = mysqli_query($enlace, $sql);
if ($datos) {
    echo "<table>";
    echo "<tr><th>ID</th><th>Nombre</th><th>IP</th><th>Tiempo de
Registro</th><th>Tiempo de Expiración</th></tr>";

    while ($fila = mysqli_fetch_assoc($datos)) {
        echo "<tr>";
        echo "<td>" . $fila['id'] . "</td>";
        echo "<td>" . $fila['nombre'] . "</td>";
        echo "<td>" . $fila['ip'] . "</td>";
        echo "<td>" . $fila['tiempo_registro'] . "</td>";
        echo "<td>" . $fila['tiempo_expiracion'] . "</td>";
        echo "<td><a href='borrar.php'> <span class='material-symbols-
outlined'> delete </span> </a></td>";

        echo "</tr>";


    }

    echo "</table>";
} else {
    echo "<p class='error'>Error al ejecutar la consulta: " .
mysqli_error($enlace) . "</p>";
}

// Cerrar la conexión a la base de datos
mysqli_close($enlace);

?>
</div>
</body>
</html>
```

El cual haremos que muestre esta pagina web el cual enseñará todas las conexiones que tiene actualmente el firewall

Usuarios Registrados					
ID	Nombre	IP	Tiempo de Registro	Tiempo de Expiración	
70	werwef@werwer	192.168.70.187	2024-05-03 19:19:18	2024-05-03 20:19:19	

Luego ahora tenemos el botón borrar el cual su función es muy simple borrar al cliente esto hará que su hora se cambie y sea la actual para así el script de Python detecte que se han pasado el tiempo y así eliminarlo.

```
<?php

session_start();
$id = $_POST['id'];
// Establecer la conexión a la base de datos
$enlace = mysqli_connect("192.168.80.20", "root", "123", "Acceso");

// Verificar si la conexión fue exitosa
if (!$enlace) {
    echo "<p class='error'>Error en la base de datos: " . mysqli_connect_error()
. "</p>";
    exit;
}

$hora = date("Y-m-d H:i:s");

$sql = "UPDATE exusuarios SET tiempo_expiracion='$hora' WHERE id=$id";
$datos = mysqli_query($enlace, $sql);
if (!$datos) {
    echo "<p class='error'>Error al ejecutar la consulta: " .
mysqli_error($enlace) . "</p>";
} else {
    echo "Registro eliminado correctamente";
}
header("Location: lista.php ");

?>
```

Podremos luego ir al cliente y así ver al usuario (usado en el buscador para server llamado Lynx)

Usuarios Registrados con Cuenta activa

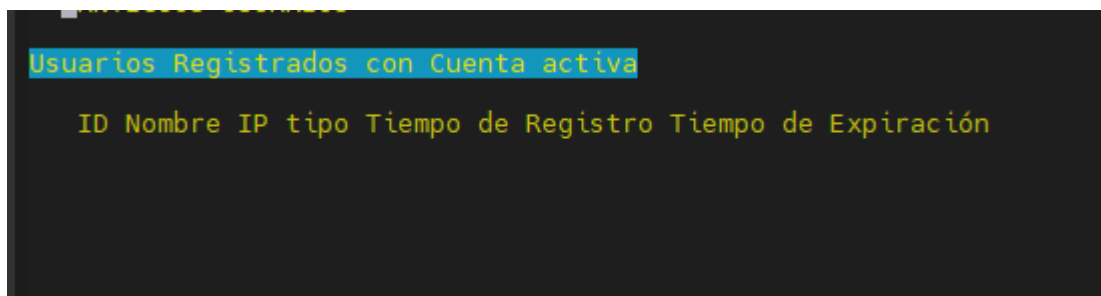
```
ID      Nombre      IP      tipo  Tiempo de Registro  Tiempo de Expiración
83 wrwerw@werrwer 192.168.70.194 gratis 2024-05-06 11:33:11 2024-05-06 12:33:11 delete
```

3.Documentacion

Y podremos ver que tenemos conexión

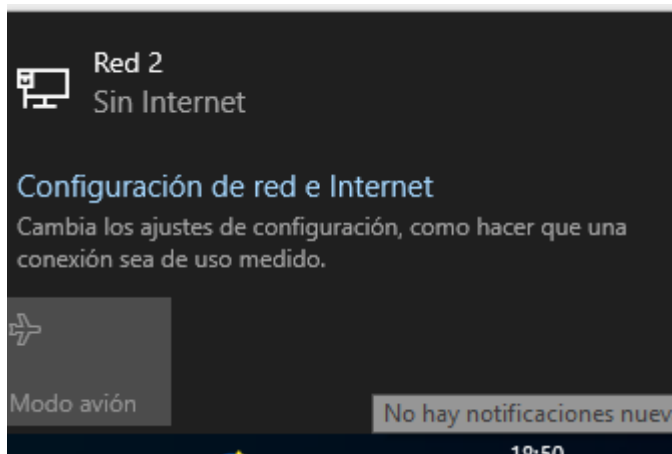


Luego podemos darle a eliminar y así se realizará el proceso de eliminación del usuario causando que el script de Python realice su proceso.



```
no hay registros
[(83, '08:00:27:b1:d2:9a', '192.168.70.194', 'gratis', 'wrwerw@werrwer', datetime.datetime(2024, 5, 6, 11, 33, 11), datetime.datetime(2024, 5, 6, 11, 52, 5))]
se acabo el tiempo
```

Y entonces lo acepta como que se acabó el tiempo y esto causa que se quede sin internet



3.Documentacion

Y acabe en la lista de antiguos usuarios, la cual se guarda para asi no dejar acceder a la gente mas de lo necesario

3	192.168.70.194	gratis	08:00:27:b1:d2:9a	2024-05-06
---	----------------	--------	-------------------	------------

Ahora tenemos que negar la conexión a los que están en la tabla antiguos_usuarios, esto se realiza con este código el cual borra la marca antigua y luego mete la nueva para que así no pueda acceder

```
borrar_marca=f"sudo {iptables} -t mangle -D PREROUTING -m mac --mac-source {mac} -j MARK --set-mark 1"
run_iptables_command(borrar_marca)
print("se acabo el tiempo")

poner_segunda=f"sudo {iptables} -t mangle -A PREROUTING -m mac --mac-source {mac} -j MARK --set-mark 2"
run_iptables_command(poner_segunda)
```

como aquí vemos se borra una y se crea otra

5	MARK	all	--	anywhere	anywhere	MAC08:00:27:ca:25:c9 MARK set 0x1
Chain INPUT (policy ACCEPT)						
4	MARK	all	--	anywhere	anywhere	MARK set 0x1
5	MARK	all	--	anywhere	anywhere	MAC08:00:27:ca:25:c9 MARK set 0x2

Luego de esto es posible que al usuario, aparecerá el portal cautivo pero tenemos un truquito gracias al ssh el cual entra a por ahí a la ip 192.168.110.170 y hace un arp de la ip que acaba de conseguir, para luego obtener la dirección mac la cual una vez la tiene, o no redireccionará a una página u otra si se encuentra la en la base de datos de antiguos usuarios.

```
$enlace=mysqli_connect("192.168.80.20", "root", "123", "Acceso");

if (!$enlace){
    echo "Error en la base de datos" . mysqli_connect_error();
    exit;
}

$host = '192.168.110.170';

$port = 22;
```

```
$username = 'helena';

$password = '123';

$ip=$_SERVER['REMOTE_ADDR'];

echo "$ip";
$connection = ssh2_connect($host, $port);

ssh2_auth_password($connection, $username, $password);
$comando='arp -n '. $ip .' | awk 'NR==2 {print $3}'';

$stream = ssh2_exec($connection, $comando);

stream_set_blocking($stream, true);

$output = stream_get_contents($stream);

$sql = "SELECT * FROM exusuarios";
$datos = mysqli_query($enlace, $sql);

while ($fila = mysqli_fetch_assoc($datos)){

    if ($fila['mac'] == $mac){
        header("Location: acabar.php");
        ssh2_disconnect($connection);
    }

}

header("Location: index2.php");
```

y podremos ver que en efecto funciona, indicando al cliente que efectivamente la sesión ha finalizado y nos agradece por ello

A screenshot of a web browser's address bar. It shows navigation icons (back, forward, refresh) on the left, a lock icon and a red 'X' icon in the middle, and the URL '192.168.70.10/acabar.php' on the right.

se le ha acabado el tiempo muchas gracias por confiar en nosotros

3.Documentacion

Luego si queremos borrar a los usuarios dentro de esto podemos, esto se podría hacer cada cierto tiempo, por si queremos liberar a los usuarios, esto lo hacemos con otra pagina la cual se asigna según la mac yendo a una pagina con el php.

```
<?php

session_start();
// Establecer la conexión a la base de datos
$mac = $_GET['mac'];

$enlace = mysqli_connect("192.168.80.20", "root", "123", "Acceso");

// Verificar si la conexión fue exitosa
if (!$enlace) {
    echo "<p class='error'>Error en la base de datos: " . mysqli_connect_error()
. "</p>";
    exit;
}

$host = '192.168.110.170';

$port = 22;

$username = 'helena';

$password = '123';

$connection = ssh2_connect($host, $port);

ssh2_auth_password($connection, $username, $password);

$stream = ssh2_exec($connection, 'sudo iptables -t mangle -D PREROUTING -m mac
--mac-source' . $mac . '-j MARK --set-mark 2');

stream_set_blocking($stream, true);

$output = stream_get_contents($stream);
print_r($output);
ssh2_disconnect($connection);

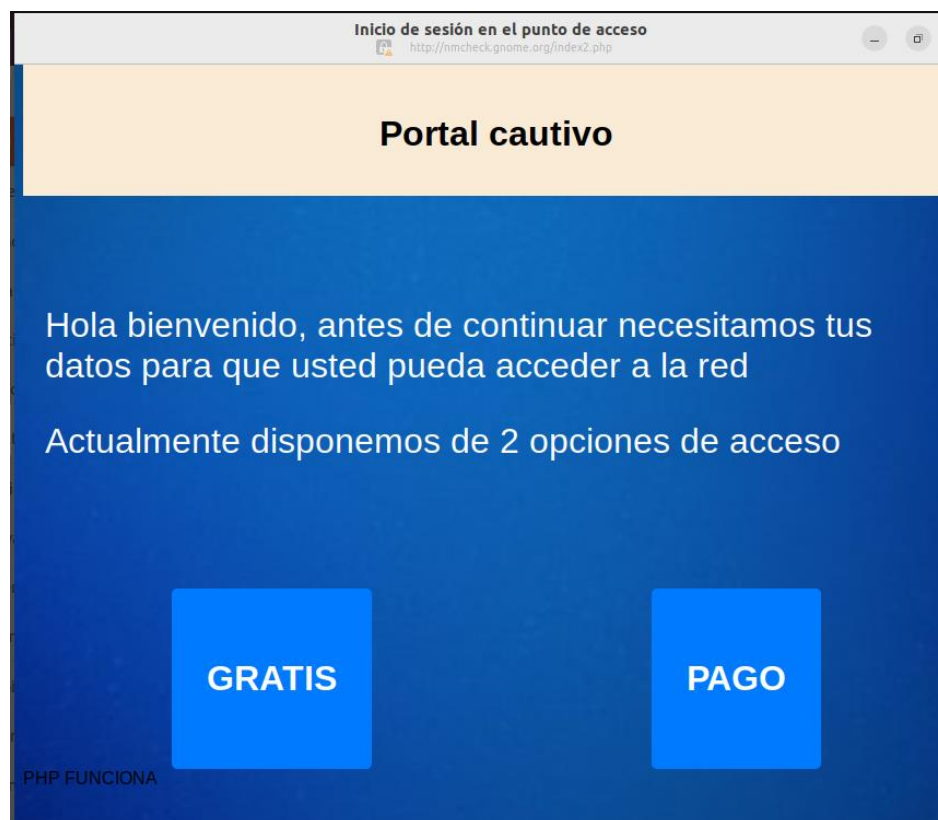
$sql = "DELETE FROM antiguos_usuarios ";
```

```
$datos = mysqli_query($enlace, $sql);

if (!$datos) {
    echo "<p class='error'>Error al ejecutar la consulta: " .
mysqli_error($enlace) . "</p>";
} else {
    header("Location: pantalla2.php ");
}

?>
```

La cual hacemos el mismo modo entrando por ssh borramos el apartado para denegar conexiones y así luego borrar al usuario de la base de datos de antiguos_usuarios el cual una vez se ejecute, redireccionará hacia la página de acceso de nuevo sin ningún problema.



3.Documentacion

3.10 Tickets

Ahora podemos crear unos tickets los cuales puede usar un usuario el cual paga para que así puedan ser compartidos con otros usuarios para que tengan un tiempo premium. Primero de todo editaremos el modo de pago para añadir el código de generación de los tokens y que se añaden

Para esto tenemos que hacer una base de datos donde los almacenamos.

```
CREATE DATABASE IF NOT EXISTS TokenDB;
```

```
USE TokenDB;
```

```
CREATE TABLE IF NOT EXISTS Tokens (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    mac_usuario VARCHAR(17),
```

```
    mac_token VARCHAR(17),
```

```
    token VARCHAR(255) ,
```

```
    tiempo_acceso DATETIME NOT NULL,
```

```
    tiempo_finalizacion DATETIME,
```

```
    nombre Text(20)
```

```
);
```

Luego introducimos el código para el modo de pago cree los tokens y los añaden

```
!reference
function random_strings($length_of_string) {
    return substr(md5(time()), 0, $length_of_string);
}

!reference
function random_strings2($length_of_string) {
    return substr(str_shuffle('0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'), 0, $length_of_string);
}

if ($tipo == 'gratis') {
    echo "Registro completado. Tienes 1 hora de tiempo. Por favor, espera unos momentos y puedes abandonar la página. Si no funciona, sal y vuelve a entrar en la red WiFi.";
} elseif ($tipo == 'pago') {
    // Genera dos tokens aleatorios para usuarios de pago
    $token1 = random_strings(8);
    $token2 = random_strings2(8);

    $insertar = "INSERT INTO Tokens (nombre, token, tiempo_acceso) VALUES ('$nom', '$token1', '$hora'), ('$nom', '$token2', '$hora')";
    $resultado2 = mysqli_query($enlace, $insertar);

    if ($resultado2) {
        echo "Registro completado. Tienes 5 horas de acceso premium. Por favor, espera unos momentos y puedes abandonar la página. Si no funciona, sal y vuelve a entrar en la red WiFi. Aquí tienes tus token";
    } else {
        echo "Registro fallido. Ha habido un fallo en el sistema. Por favor, informa a uno de nuestros tenderos, ellos te ayudarán.";
    }
} else {
    echo "Tipo de conexión desconocido.";
}

mysqli_close($enlace);
exit;
```

3.Documentacion

Registro completado. Tienes 5 horas de acceso premium. Por favor, espera unos momentos y puedes abandonar la página. Si no funciona, sal y vuelve a entrar en la red WiFi. Aquí tienes tus tokens: bb0b29eb, v8HRIbmK

Y vemos que se nos añade a la base de datos:

6	NULL	NULL	ac4cc7b7	2024-05-07 05:16:37	NULL	sadasds
7	NULL	NULL	bb0b29eb	2024-05-07 05:21:31	NULL	sadasds
8	NULL	NULL	v8HR1bmK	2024-05-07 05:21:31	NULL	sadasds

Ahora añadimos código a la pagina index para asi añadir la opción de los tickets

```
</div>
<div class="cubo2">

    <a class="color" href="./ticket.php"><p>Ticket</p></a>

</div>
```

Para luego añadir una pagina dedicada exclusivamente a estas

```

<html>
<lang="es">
</>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registro de Usuario</title>
<link rel="stylesheet" href="Css/Registro.css">
</>
<div class="cuadrado">
<h2>Canjeado de Tickets</h2>
</div>
<form action="process_ticket.php" method="GET">
  <label for="nombre">Ticket</label>
  <input type="text" id="ticket" name="ticket" required>

  <input type="hidden" id="tipo" name="tipo" value="ticket"> </input>
  <label> <input type="checkbox" id="checkboxbox" name="checkboxbox" required> <a href="condici

  <input type="submit" value="Canjear" >

</form>
</>
</>

```

El cual su función solo es que se introduzca el ticket y se acepten los términos y condiciones luego tenemos que hacer que lo busque si existe o no entonces haremos una pagina de proceso la cual procesara.

Ahora luego tenemos que hacer que se introduzcan los datos dentro de la base de datos

```
<?php
session_start();

$ticket = $_POST['tic'];
```

```

$enlace = mysqli_connect("localhost", "root", "123", "Acceso");

if (!$enlace) {
    echo "Error en la base de datos: " . mysqli_connect_error();
    exit;
}

$ip = $_SERVER['REMOTE_ADDR'];
$hora = date("Y-m-d H:i:s");
$tipo = "ticket";
$nom = "na";
$correo = "na";

$codigo = "SELECT * FROM Tokens";
$datos = mysqli_query($enlace, $codigo);

if (!$datos) {
    echo "Error al ejecutar la consulta: " . mysqli_error($enlace);
    exit;
}

$ticket_encontrado = false;

while ($fila = mysqli_fetch_assoc($datos)) {
    echo $fila['token'];
    echo " ";
    echo $ticket;
    if ($fila['token'] == $ticket) {
        $ticket_encontrado = true;

        break;
    }
}

if ($ticket_encontrado) {
    $insertar = "INSERT INTO registro_conexiones (nombre, correo, ip, duracion,
tipo, fecha) VALUES ('$nom', '$correo', '$ip', NULL, '$tipo', '$hora')";
    $resultado = mysqli_query($enlace, $insertar);

    if (!$resultado) {
        echo "Error al insertar registro de conexión: " . mysqli_error($enlace);
        exit;
    }

    $delete = "DELETE FROM Tokens WHERE token = '$ticket'";
    $resultado = mysqli_query($enlace, $delete);

    if (!$resultado) {

```

3.Documentacion

```
        echo "Error al eliminar el ticket: " . mysqli_error($enlace);
        exit;
    }

    echo "Registro completado, tienes 5 horas de tiempo. Recuerda, espera unos
momentos y luego puedes abandonar la página. Si no funciona, sal y vuelve a entrar
en el WiFi.";
} else {
    echo "El ticket proporcionado no es válido.";
}

mysqli_close($enlace);
#header("Location: index2.php");
exit;
?>
```

entonces usaremos este código para introducir los datos, luego se procesará como otro mas como si fuera uno de pago o uno gratis en el script de Python.

```
hora_registro = datetime.now()
if tipo=="gratis":
    una_hora = timedelta(hours=1)
    tiempo_exp= hora_registro + una_hora

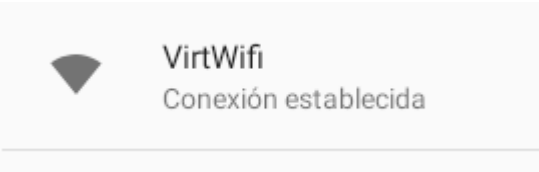
elif tipo=="pago":
    una_hora = timedelta(hours=5)
    tiempo_exp = hora_registro + una_hora
elif tipo=="ticket":
    una_hora = timedelta(hours=5)
    tiempo_exp = hora_registro + una_hora

consulta = "INSERT INTO exusuarios (mac, ip, tipo, nombre,
tiempo_registro, tiempo_expiracion) VALUES (%s, %s, %s, %s, %s, %s)"
datos = (mac, ip, tipo, nombre, hora_registro, tiempo_exp)
cursor1.execute(consulta, datos)
conexion1.commit()
conexion1.close()
```

3.Documentacion

y podremos ver que ocurre y que tenemos internet

```
tetime(2024, 5, 7, 21, 36, 41), datetime.datetime(2024, 5, 8, 2, 36, 41)), (93,
'08:00:27:cb:29:62', '192.168.70.196', 'ticket', 'na', datetime.datetime(2024, 5
, 7, 22, 0, 50), datetime.datetime(2024, 5, 8, 3, 0, 50))]
no hay registros
```



Y al rato caducará el código.

Luego también se puede hacer una pagina donde están los tickets.

TOKENS DE USUARIOS

ID	token	nombre	
1	3fb3ffed	sadasds	
2	3fb3ffed	sadasds	
3	5363a439	sadasds	
4	5363a439	sadasds	
7	bb0b29eb	sadasds	
8	v8HRlbmK	sadasds	
9	be3df6bd	qwewqege	
10	PthB6pOy	qwewqege	
11	6ae84080	erwrwewe	

3.11 Velocidad

Ahora añadiremos la velocidad para así dependiendo el tipo sea usado por uno o por otro tipo, esto será gracias a tc o también llamado traffic control.

Este sistema es usado para limitar el ancho de banda de los dispositivos conectados a una red el cual maneja tanto la latencia la fiabilidad y el costo.

Esto lo instalando con

`Sudo apt-get install tc`

Y luego añadimos estos comandos:

`Sudo tc qdisc add dev enp0s8 root`

`Sudo tc qdisc add dev enp0s8 root handle 1: htb default 11`

PAGO

`Sudo tc class add dev enp0s8 parent 1: classid 1:10 htb rate 2mbit ceil 2mbit`

GRATIS:

`Sudo tc class add dev enp0s8 parent 1: classid 1:11 htb rate 250kbit ceil 250kbit`

Ahora le indicamos al servidor que queremos hacer esto

```
helenahelena:/etc/iptables$ sudo tc qdisc add dev enp0s8 root handle 1: htb default 11
helenahelena:/etc/iptables$ sudo tc class add dev enp0s8 parent 1: classid 1:10 htb rate 2mbit c
eil 2mbit
helenahelena:/etc/iptables$ sudo tc class add dev enp0s8 parent 1: classid 1:11 htb rate 250kbi
t ceil 250kbit
helenahelena:/etc/iptables$ █
```

`Sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst 192.168.10.101 flowid 1:11`

`Sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst 192.168.10.102 flowid 1:10`

Añadimos dentro del ejecutable de Python para que haga esto

3.Documentacion

```
hora_registro = datetime.now()
if tipo=="gratis":
    una_hora = timedelta(hours=1)
    tiempo_exp= hora_registro + una_hora
    comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst {ip} flowid 1:11"
    print("se ha insertado gratis")
elif tipo=="pago":
    una_hora = timedelta(hours=5)
    comavel=f" tc filter add dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst {ip} flowid 1:10"
    print("se ha insertado pago")
    tiempo_exp = hora_registro + una_hora
elif tipo=="ticket":
    una_hora = timedelta(hours=5)
    tiempo_exp = hora_registro + una_hora
    print("se ha insertado ticket")
    comavel=f"tc filter add dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst {ip} flowid 1:10"

run_iptables_command(comavel)
```

una vez ejecutado funcionará

luego si queremos borrar la norma a esa ip entonces añadiremos unas líneas en la parte de borrar que según el tipo y junto a la ip entonces se borrará la línea que hay ligada a la ip que esta en uso

```
if tipo=="gratis":
    comavel=f"sudo tc filter del dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst {ip} flowid 1:10"

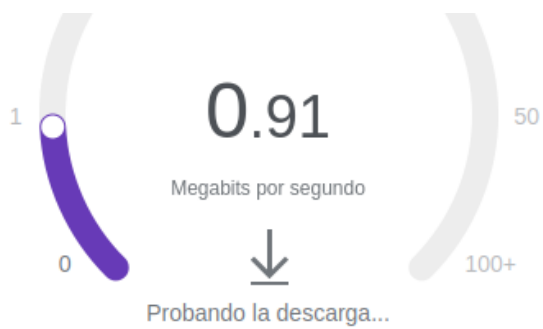
elif tipo=="pago":
    comavel=f" tc filter del dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst {ip} flowid 1:11"

elif tipo=="ticket":
    comavel=f"tc filter del dev enp0s8 protocol ip parent 1:0 prio 1 u32 match ip dst {ip} flowid 1:11"

run_iptables_command(comavel)
```

Ahora hacemos el test del modo gratis el cual tendrá ya 1mb:

3.Documentacion



El cual tendrá una velocidad muy baja pero suficiente para repartirlo al resto

Y ahora el premium con 5mb:



3.12 SSL a la pagina web

Vamos a meter ssl a la pagina web para que la introducción de datos no sea vista por los hackers o gente chismosa que quiere tus datos.

Para ello tendremos que hacer `a2enmod ssl` y luego reiniciar el server.

```
helena@helena:/var/www/html$ sudo a2enmod ssl
[sudo] password for helena:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
```

Ahora creamos la carpeta donde estarán los certificados

```
helenahelena:/var/www/html$ sudo mkdir /etc/apache2/cert
helenahelena:/var/www/html$
```

Y luego vamos a crear el certificado auto firmado para que el ssl lo use

```
helena@helena:/var/www/html$ sudo make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/cert/apache.pem
helena@helena:/var/www/html$
```

Ahora vamos a la pagina web y lo editamos

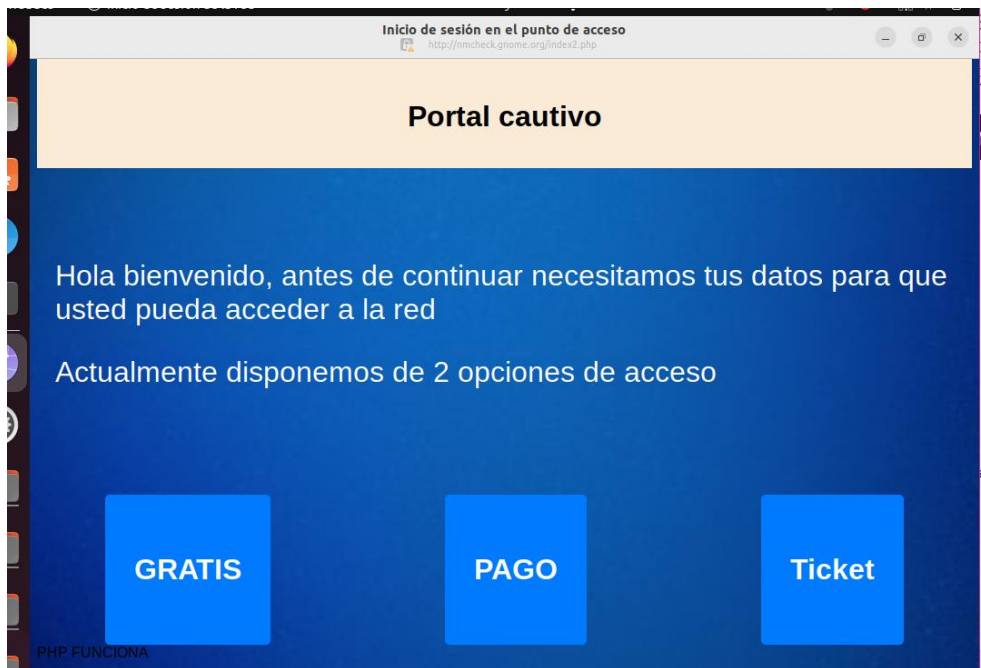
```
helena@helena: /var/www/html$ sudo nano /etc/apache2/sites-enabled/000-default.conf
helena@helena: /var/www/html$ sudo nano /etc/apache2/sites-enabled/000-default.conf
f
```

Y añadimos esto

```
GNU nano 6.2 /etc
<VirtualHost *:443>
    # The ServerName
    # the server uses
    SSLEngine on
    SSLCertificateFile /etc/apache2/certs/apache.pem
    SSLProtocol all
    SSLCipherSuite HIGH:MEDIUM
```

3.Documentacion

Y luego reiniciamos el servicio y probamos



Aun que al parecer no acepta ssl por desgracia.

3.13 Todo unido

Ahora nuestro objetivo será unirlo todo, al inicio tenemos que introducir que se ejecute el archivo de iptables, primero que borre todos los registros y luego los cree.

```
archivo= open("./limpia.sh")
lineas = archivo.readlines()
print(lineas)
for linea in lineas:
    if linea != "\n":
        texto= "sudo " + linea
        print(texto)

        correr(texto)
logger.info('SE BORRO TODO IPTABLES')
archivo.close()

correr ("echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward")

try:
    logger.info('se ejecuto EL ARCHIVO DE IPTABLES')
    archivo2= open("./tablas.sh")
    lineas2 = archivo2.readlines()
    for linea2 in lineas2:

        if linea2 != "\n":
            texto2= "sudo " + linea2
            print(texto2)
            correr(texto2)
```

Luego podemos indicar cuales son los parametros del TRAFIC CONTROL, según las velocidades que tiene cada uno

```
try:
    correr(borrar_todo)
except:
    print("no existe ya")
    logger.info('YA SE BORRO EL TRAFIC CONTROL AL INICIO')

try:
    run="sudo tc qdisc add dev enp0s8 root"
    correr(run)
```

3.Documentacion

```
logger.info('NO SE PUDO ESTABLECER LOS PARAMETROS DEL TRAFIC CONTROL AL
INICIO')
except:
    print("no se pudo")

run2="sudo tc qdisc add dev enp0s8 root handle 1: htb default 11"
correr(run2)

run3="sudo tc class add dev enp0s8 parent 1: classid 1:10 htb rate 5mbit ceil 5mbit"
correr(run3)

run4="sudo tc class add dev enp0s8 parent 1: classid 1:11 htb rate 1mbit ceil 1mbit"
correr(run4)
```

Luego podemos ver si en la base de datos de exusuarios hay alguien y según su tipo de se le añade una regla de nuevo según su ip

```
conexion1 = mysql.connector.connect(host="192.168.80.20", user="root", passwd="123")
cursor1 = conexion1.cursor()
cursor1.execute("USE Acceso")

cursor1.execute("SELECT * FROM exusuarios")

resulta2 = cursor1.fetchall()
if resulta2:
    for inicio in resulta2:
        tipo=inicio[6]
        ip = inicio[3]

        if tipo=="gratis":

            comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1
u32 match ip dst {ip} flowid 1:11"
            print("se ha insertado gratis")
        elif tipo=="pago":

            comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1
u32 match ip dst {ip} flowid 1:10"
            print("se ha insertado pago")

        elif tipo=="ticket":

            print("se ha insertado ticket")
```

3.Documentacion

```
comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1
u32 match ip dst {ip} flowid 1:10"
```

y luego añadimos que según la base de datos de exusuarios se añada su configuración de iptables de nuevo, como si nada hubiera pasado

```
conexion1 = mysql.connector.connect(host="192.168.80.20", user="root", passwd="123")
cursor1 = conexion1.cursor()
cursor1.execute("USE Acceso")

cursor1.execute("SELECT * FROM exusuarios")

resulta2 = cursor1.fetchall()
if resulta2:
    for inicio in resulta2:
        tipo=inicio[3]
        ip = inicio[2]
        mac=inicio[1]

        if tipo=="gratis":

            comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1
u32 match ip dst {ip} flowid 1:11"
            print("se ha insertado gratis")
            elif tipo=="pago":

                comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1
u32 match ip dst {ip} flowid 1:10"
                print("se ha insertado pago")

            elif tipo=="ticket":

                print("se ha insertado ticket")
                comavel=f"sudo tc filter add dev enp0s8 protocol ip parent 1:0 prio 1
u32 match ip dst {ip} flowid 1:10"

                run_iptables_command(comavel)
                coma=f"sudo {iptables} -t mangle -A PREROUTING -m mac --mac-source {mac} -j
MARK --set-mark 1"
                run_iptables_command(coma)
```

3.Documentacion

VIDEO DEL FUNCIONAMIENTO: <https://youtu.be/qBdH35NnqLI>

3.14 Usuarios Bloqueados (Extra)

Ahora tenemos una base de datos completa, pero si vemos a alguien que hace algo mal? O un mal uso de esta? No te preocupes vamos a añadir la base de datos de usuarios bloqueados y una opción para así verlos.

```
CREATE TABLE usuarios_bloqueados (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre_usuario VARCHAR(50),  
  correo_electronico VARCHAR(100),  
  razon_bloqueo TEXT NOT NULL,  
  mac_bloqueada TEXT not null);
```

Ahora deberemos hacer la pagina

```
<?php  
  
session_start();  
if(!$_SESSION['registrado']){  
    //encabezado de redirección  
    header("location:lista.php");  
    die;  
}  
// Establecer la conexión a la base de datos  
$enlace = mysqli_connect("192.168.80.20", "root", "123", "Acceso");  
  
// Verificar si la conexión fue exitosa  
if (!$enlace) {  
    echo "<p class='error'>Error en la base de datos: " . mysqli_connect_error()  
    . "</p>";  
    exit;  
}  
  
?>  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <link rel="stylesheet" href="Css/tabla.css">  
    <link rel="stylesheet"  
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,F  
ILL,GRAD@20..48,100..700,0..1,-50..200" />  
    <title>Usuarios Registrados</title>
```

```

</head>
<body>
<a class="button" href='pantalla2.php'>ANTIGUOS USUARIOS </a>
<a class="button" href='pantalla.php'> USUARIOS CONECTADOS </a>
<a class="button" href='pantalla2.php'>Exusuarios logeados</a>
<a class="button" href='salir.php'>Cerrar Session</a>
    <div class="container">

        <h1>Usuarios Bloqueados</h1>

        <?php

            $sql = "SELECT * FROM usuarios_bloqueados";
            $datos = mysqli_query($enlace, $sql);

            if ($datos) {
                echo "<table>";
                echo
" <tr><th>ID</th><th>Nombre</th><th>Correo</th><th>Razon</th><th>TMac</th><th>Desbloq
uear</th>";

                while ($fila = mysqli_fetch_assoc($datos)) {
                    echo "<tr>";
                    echo "<td>" . $fila['id'] . "</td>";
                    echo "<td>" . $fila['nombre_usuario'] . "</td>";
                    echo "<td>" . $fila['correo_electronico'] . "</td>";
                    echo "<td>" . $fila['razon_bloqueo'] . "</td>";
                    echo "<td>" . $fila['mac_bloqueada'] . "</td>";
                    echo '<td><a href="desbloquear.php?id=' . $fila['id'] . '"> <span
class="material-symbols-outlined"> delete </span> </a></td>';

                    echo "</tr>";

                }

                echo "</table>";
            } else {
                echo "<p class='error'>Error al ejecutar la consulta: " .
mysqli_error($enlace) . "</p>";
            }

```



```



        mysqli_close($enlace);
    ?>
</div>
</body>
</html>

```

Usuarios Bloqueados

ID	Nombre	Correo	Razon	TMac	Desbloquear
----	--------	--------	-------	------	-------------

Luego tenemos que hacer que en alguna pagina funcione el bloqueo de usuarios en este caso será la de antiguos usuarios se añade una indicación para bloquear el usuario.

ID	ip	nombre	MAC	Fecha de expiración		
38	192.168.70.197	gratis	08:00:27:ca:25:c9	2024-05-15		

Si le damos nos redirige a usuarios bloqueados este es el código completo de lo que hace

```

<?php
session_start();

$id = $_GET['id'];
$mac = $_GET['mac'];

// Definir variables adicionales necesarias
$nombre = "Usuario"; // Reemplazar con el valor adecuado si está disponible

$host = '192.168.80.10';
$port = 22;
$username = 'helena';
$password = '123';

```

```

// Establecer la conexión a la base de datos
$enlace = mysqli_connect("192.168.80.20", "root", "123", "Acceso");
if (!$enlace) {
    echo "Error en la base de datos: " . mysqli_connect_error();
    exit;
}

// Establecer la conexión SSH
$connection = ssh2_connect($host, $port);
if (!$connection) {
    echo "<p class='error'>Error al conectar por SSH al servidor remoto</p>";
    exit;
}

if (!ssh2_auth_password($connection, $username, $password)) {
    echo "<p class='error'>Error de autenticación SSH</p>";
    exit;
}

$stream = ssh2_exec($connection, "sudo iptables -t mangle -D PREROUTING -m mac --
mac-source $mac -j MARK --set-mark 2");
stream_set_blocking($stream, true);
$output = stream_get_contents($stream);


// Cerrar la conexión SSH
ssh2_disconnect($connection);

// Eliminar la entrada de la base de datos
$sql = "DELETE FROM antiguos_usuarios WHERE mac = '$mac'";
$datos = mysqli_query($enlace, $sql);


if (!$datos) {
    echo "<p class='error'>Error al ejecutar la consulta: " . mysqli_error($enlace)
    . "</p>";
} else {
    echo "pito";
    $insertar2 = "INSERT INTO usuarios_bloqueados (mac_bloqueada, nombre_usuario,
correo_electronico,razon_bloqueo) VALUE ('$mac',NULL, NULL,'NOC')";
    $resultado2 = mysqli_query($enlace, $insertar2);
    echo "muac";
    if (!$resultado2) {
        echo "<p class='error'>Error al ejecutar la consulta: " .
mysqli_error($enlace) . "</p>";
    } else {
        header("Location: usuariosbloqueados.php");
        exit;
    }
}

```

```
}  
  
// Cerrar la conexión a la base de datos  
mysqli_close($enlace);  
?>
```

ID	Nombre	Correo	Razon	TMac	Desbloquear
2			NOC	08:00:27:ca:25:c9	

Ahora podremos ver el usuario que esta bloqueado y efectivamente no puede entrar y nos saldrá esta indicación en el portal cautivo

 <http://nmcheck.gnome.org/bloq.php?mac=08:00:27:ca:25:c9>

HA SIDO BLOQUEADO HABLA CON EL ADMINISTRADOR DEL SISTEMA

Y no podremos acceder a internet

Luego de ello si lo queremos desbloquear podremos hacerlo aquí en el botón desbloquear, la cual solo lo borrará del registro ahora si permitiéndole entrar de nuevo.

3.15 Servidor DNS + Proxy

Ahora que me ha dado tiempo haremos el servidor dns esto se hace:

Editamos el archivo forwarders

```
forwarders
{
8.8.8.8;
};
```

Hacemos el archivo de delimitación de las zonas

```
zone "xarxa.lan" {
    type master;
    file "/etc/bind/direct_xarxa";
};

zone "110.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/inversa_xarxa";
};
```

Creamos los ficheros inversos y directos

ZONA DIRECTA
<pre>; ZONA DIRECTA DE L'AULA ; \$TTL 86400 @ IN SOA xarxa.lan. root.xarxa.lan. (1 ; Serial 604800 ; Refresh 86400 ; Retry 2419200 ; Expire 86400) ; Negative Cache TTL ; @ IN NS xarxa.lan. xarxa.lan. IN A 192.168.0.1</pre>
ZONA INVERSA

3.Documentacion

```
; ZONA INVERSA DE L'AULA
;
$TTL      86400
@         IN      SOA     xarxa.lan. root.xarxa.lan. (
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        86400 ) ; Negative Cache TTL
;
@         IN      NS      xarxa.lan.

1         IN      PTR     xarxa.lan.
```

Y reiniciamos, luego editamos el servidor DHCP

```
option domain-name "xarxa.lan";
default-lease-time 6000;
max-lease-time 7200;
option subnet-mask 255.255.255.0;

subnet 192.168.70.0 netmask 255.255.255.0 {
    range 192.168.70.60 192.168.70.200;
    option routers 192.168.70.10;
    option broadcast-address 192.168.70.255;
    option domain-name-servers 192.168.110.170;
}
default-lease-time 600;
max-lease-time 7200;
```

Finalmente lo reiniciamos

Y luego editamos el archivo de iptables y le añadimos algunas cosas con que el servidor DNS es cierta ip

Y que la ip tenga permiso de usar los DNS

3.Documentacion

```
/sbin/iptables -A OUTPUT -d 192.168.110.170 -p udp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -s 192.168.110.170 -p udp --sport 53 -m state --state
RELATED,ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -d 192.168.110.170 -p tcp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -s 192.168.110.170 -p tcp --sport 53 -m state --state
RELATED,ESTABLISHED -j ACCEPT

/sbin/iptables -A OUTPUT -d 8.8.8.8 -p udp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -s 8.8.8.8 -p udp --sport 53 -m state --state
RELATED,ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -d 8.8.8.8 -p tcp --dport 53 -j ACCEPT
/sbin/iptables -A INPUT -s 8.8.8.8 -p tcp --sport 53 -m state --state
RELATED,ESTABLISHED -j ACCEPT
```

Ahora tenemos que instalar el proxy esta vez lo haremos con un servidor el cual pueda acceder pero primero necesitamos un sistema de certificación esto se hará con el openssl

```
helena@helena:/etc/iptables$ sudo apt install openssl libssl-dev
```

Vamos al archivo /etc/ssl/openssl.cnf

```
dir                = /etc/ssl/caijm                # Where everything is kept
certs              = $dir/certs                    # Where the issued certs are kept
crl_dir            = $dir/crl                      # Where the issued crl are kept
database           = $dir/index.txt                # database index file.
#unique_subject    = no                           # Set to 'no' to allow creation of
# several certs with same subject.
new_certs_dir      = $dir/newcerts                 # default place for new certs.

certificate        = $dir/caijm.crt                # The CA certificate
serial            = $dir/serial                    # The current serial number
crlnumber          = $dir/crlnumber                # the current crl number
# must be commented out to leave a V1 CRL
crl               = $dir/crl.pem                  # The current CRL
private_key        = $dir/private/caijm.key        # The private key
```

Y creamos la estructura de archivos

```
mkdir /etc/ssl/caijm/
mkdir /etc/ssl/caijm/certs
mkdir /etc/ssl/caijm/private
mkdir /etc/ssl/caijm/newcerts
mkdir /etc/ssl/caijm/crl
echo "01" > /etc/ssl/caijm/serial
touch /etc/ssl/caijm/index.txt
```

3.Documentacion

y luego creamos el fichero de certificación

```
cd /etc/ssl/caijm/  
openssl req -nodes -new -x509 -keyout private/caijm.key -out caijm.crt -days 3650
```

```

eq: can't open private/caijm.key for writing, Permission denied
elena@elena:/etc/ssl/caijm$ sudo openssl req -nodes -new -x509 -keyout private/caijm.key -out private/caijm.crt -days 3650
+++++*+...+.....+...+
..+..+.....+...+.....+...+++++
+++++*+...+.....+...+.....+...+
      +      +      +      +      +      +      +      +      +      +

```

Luego nos instalamos un fichero y lo descomprimos

```
helenahelena:/etc/ssl/caijm$ sudo tar -xf /home/helena/squid-5.9.tar.gz
helenahelena:/etc/ssl/caijm$
```

Y hacemos este comando dentro del fichero

```
carriera:carriera.sh evltt@inter example.log images.jpeg squid-5.9 squid-5.9.tar.gz
helena@helena:~$ cd squid-5.9/
helena@helena:~/squid-5.9$ ./configure --enable-icap-client --enable-ssl --enable-ssl-crtld --with-openssl
```

Y luego ejecutamos el comando make y después de como 1 hora

Lo conseguiremos y tendremos que hacer un make install

```
make: *** [Makefile:394: install-recursive] Error 1
helena@helena:~/squid-5.9$ sudo make install
```

Ahora vamos a

```
helena@helena:~/squid-5.9$ sudo nano /usr/local/squid/etc/squid.conf
```

Y hacemos un copypaste de esto

```
http port 192.168.0.1:3128
```

```
http port 192.168.0.1:3129 intercept
```

```
https port 192.168.0.1:3130 intercept ssl-bump generate-host-certificates=on \
```

```
dynamic cert mem cache size=4MB cert=/etc/ssl/caijm/caijm.crt \
```

```
key=/etc/ssl/caijm/private/caijm.key
```

```
cache_dir ufs /usr/local/squid/var/cache/squid 1500 16 256
```

3.Documentacion

```
always_direct allow all
ssl_bump none localhost
ssl_bump client-first all
sslproxy_cert_error allow all
```

```
acl totes src all
acl xarxa src 192.168.0.0/24
acl prohibit dstdom_regex agora.xtec.cat bbva.es
acl noxs url_regex facebook
```

```
http_access allow localhost
http_access allow xarxa !prohibit !noxs
http_access deny totes
```

```
debug_options ALL,2
```

creamos este fichero

```
helena@helena:~/squid-5.9$ touch /usr/local/squid/var/logs/cache.log
```

Hacemos el chown de la carpeta

```
helena@helena:~/squid-5.9$ sudo touch /usr/local/squid/var/logs/cache.log
helena@helena:~/squid-5.9$ chown -R proxy:proxy /usr/local/squid/var/logs
```

Luego le damos permiso de nuevo

```
ory
helena@helena:~/squid-5.9$ sudo chown -R proxy:proxy /usr/local/squid/var/cache/
squid
```

Y luego creamos este fichero

```
mkdir /usr/local/squid/var/lib
```

```
ory
helena@helena:~/squid-5.9$ sudo mkdir /usr/local/squid/var/lib
helena@helena:~/squid-5.9$
```

Creamos la carpeta de los certificados

```
chown -R proxy:proxy /usr/local/squid/var/cache/squid/ssl_db
```

y luego vamos al archivo /etc/passwd y vamos a proxy a darle permisos de /bin/bash

3.Documentacion

```
proxy:x:13:13:proxy:/bin:/bin/bash
```

Luego le damos permiso a el usuario

```
chown -R proxy:proxy /usr/local/squid/var/run
```

damos permiso para que se use el usuario

```
helena@helena:~/squid-5.9$ sudo chmod -R 755 /etc/ssl/caijm/private/
```

Creamos las carpetas del usuario

```
su proxy -c "/usr/local/squid/sbin/squid -z"
```

```
helena@helena:~/squid-5.9$ sudo su proxy -c "/usr/local/squid/sbin/squid -z"
helena@helena:~/squid-5.9$
```

Y luego ejecutamos este comando para activarlo

```
su proxy -c "/usr/local/squid/sbin/squid -NCd1".
```

Luego al acceder nos dará un error de autenticación



No se ha conectado: Posible problema de seguridad

Firefox ha detectado una potencial amenaza de seguridad y no ha continuado a **www.youtube.com** porque este sitio web requiere una conexión segura.

¿Qué puede hacer al respecto?

www.youtube.com tiene una política de seguridad llamada HTTP Strict Transport Security (HSTS), que significa que Firefox solo puede conectarse a él de forma segura. No puede añadir una excepción para visitar este sitio.

El problema está probablemente en el sitio web, y no hay nada que pueda hacer para resolverlo.

Si está en una red corporativa o utilizando un antivirus, puede ponerse en contacto con el equipo de asistencia para obtener ayuda. También puede notificar el problema al administrador del sitio web.

[Más información...](#)

Ir atrás

Avanzado...

Luego metemos la certificación y funcionaria

3.Documentacion

```
helenahelena:/etc/iptables$ sudo scp /etc/ssl/caijm/caijm.crt alumne@192.168.70.197:/home/alumne
The authenticity of host '192.168.70.197 (192.168.70.197)' can't be established.
ED25519 key fingerprint is SHA256:Nf4rXXtysJzT63vnp0vFcEyScwzb0A15+UW2qGVc9jI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.70.197' (ED25519) to the list of known hosts.
alumne@192.168.70.197's password:
caijm.crt                               100% 1245    70.3KB/s   00:00
helenahelena:/etc/iptables$ sudo su proxy -c "/usr/local/squid/sbin/squid -NCd1
"
```



Funciona pero es demasiado molesto.

FALLOS Y COMO RESOLVERLOS

1.Fechas

A veces se quedan pilladas las fechas si se pausan durante un cierto tiempo, ahora como resolverlo

Esto se resuelve reiniciando el servidor

O poner esto:

```
Sudo ntpdate 0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org 3.pool.ntp.org 0.south-america.pool.ntp.org
```

2.Ngrix Suplanta apache

Apache 2 no funciona y eso es por que ngrix funciona antes que apache2 como vemos en la imagen

3.Documentacion

```
[sudo] password for helena:
x apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Thu 2024-05-09 16:31:03 CEST; 23min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 615 ExecStart=/usr/sbin/apachectl start (code=exited, status=1/FAILURE)
      CPU: 145ms
```

Lo que hay que hacer es parar nginx o borrarlo y luego activar apache

```
helena@helena:~$ sudo systemctl stop nginx.service
```

Y ahora funcionará de nuevo

```
helena@helena:~$ sudo systemctl restart apache2
helena@helena:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-05-09 16:56:30 CEST; 2s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2291 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 2295 (apache2)
    Tasks: 6 (limit: 4031)
   Memory: 12.6M
      CPU: 107ms
   CGroup: /system.slice/apache2.service
           └─2295 /usr/sbin/apache2 -k start
             └─2296 /usr/sbin/apache2 -k start
               └─2297 /usr/sbin/apache2 -k start
                 └─2298 /usr/sbin/apache2 -k start
                   └─2299 /usr/sbin/apache2 -k start
                     └─2300 /usr/sbin/apache2 -k start
```

3.No se ejecuta el portal cautivo

Quita la interfaz 2 del server web , esto a veces pasa por que se queda pillado , desactívalo y actívalo de nuevo y funcionará.

4.Aplicabilidad

4.1 Propuesta de valor

El proyecto se centra en la seguridad de una red publica de tener el registro de los usuarios, de lo que entra y de lo que sale, para luego el administrador pueda tener acceso a eso para asi obtener una vista de lo que visitan los usuarios que están haciendo uso de la red, Entre otras cosas.

4.2 .Objetivo

El objetivo es muy simple dar acceso a los usuarios de una red mediante el portal cautivo que relaizará los siguientes puntos:

- Autenticación de Usuarios: El portal cautivo puede requerir que los usuarios ingresen sus datos y como acceso temporal (como un nombre de usuario y una contraseña) o acepten los términos y condiciones antes de poder acceder a la red.
- Control de Acceso: Esto permite a los administradores de red tener un control sobre quién puede acceder a la red y cuándo.
- Implementación de Políticas de Seguridad: Además, se puede utilizar para aplicar políticas de seguridad estas podrían ser, como el filtrado de contenido, el control de acceso basado en roles y la encriptación de datos, entre otras medidas.
- Monitoreo y Registro: También proporciona la capacidad de monitorear y registrar la actividad de los usuarios en la red.
- Publicidad y Marketing: Los portales cautivos también pueden ser utilizados para obtener datos de los usuarios que visitan la red.

4.3.Usuario y Cliente final

Tenemos en este caso 2 tipos de cliente potencial el cual actúan:

1.Usuario administrador de red

- Este usuario se encarga de configurar y administrar el portal cautivo.
- Sus funciones pueden incluir establecer políticas de acceso, autenticación de usuarios, gestionar cuentas, monitorear la actividad de la red y solucionar problemas técnicos.
- Tienen acceso privilegiado a la configuración y los registros del sistema.
- Pueden establecer reglas de seguridad, como filtrado de contenido, control de ancho de banda y restricciones de acceso basadas en horarios o ubicaciones.
- Su objetivo principal es garantizar la seguridad y eficiencia de la red, así como brindar una experiencia de usuario fluida y segura.

2. Usuario que usa los servicios

- Este usuario es cualquier persona que intenta acceder a la red Wi-Fi protegida por el portal cautivo.
- Su objetivo principal es obtener acceso a la red para utilizar los servicios disponibles, como navegar por Internet, acceder a recursos compartidos, enviar correos electrónicos, etc.
- Deben pasar por el proceso de autenticación proporcionando credenciales o aceptando los términos y condiciones del servicio antes de poder acceder completamente a la red.
- La experiencia del usuario debe ser buena para este tipo de usuario, ya que un proceso de autenticación sin problemas y una navegación fluida pueden afectar su percepción del servicio y su disposición a volver a utilizarlo en el futuro.

4.Aplicabilidad

4.4 Usabilidad

Provee servicio de internet al usuario a cambio de que los usuarios den sus datos de navegación. El cual tenga accesibilidad sea un sistema visualmente bonito y entendible para el usuario además luego sea compatible con varios sistemas operativos, como Android, Windows, Linux. El sistema también tiene que ser rápido y que haga eficientemente el registro de los usuarios y el de los datos de estos.

4.5 Intermediario / Comprador / Promotor

Esto puede interesarle a una proveedora de internet, ya sea como Jazztel, Telefónica, Yoigo entre otros, para aplicarlo en sitios grandes como un restaurante o un aeropuerto ya que usa la tecnología que vende en este caso internet.

4.6 Puntos fuertes y débiles

Puntos fuertes:

Sistema simple: Es un sistema que requiere muy pocos recursos por lo que no se necesita una maquina demasiado potente para ponerlo en marcha.

Control de acceso: Un portal cautivo te permite controlar quién tiene acceso a la red, y bloquear a los usuarios que están haciendo uso indebido de esta misma evitando así cualquier uso indebido o malicioso de la misma.

Recopilación de datos: Te brinda la posibilidad de obtener un mayor conocimiento sobre los diferentes perfiles o personas que visitan tu local.

Eficiencia de la red: Permite limitar el acceso a ciertos sitios web, lo que puede mejorar la eficiencia de la red.

Puntos débiles:

Seguridad: Si se configura de manera insegura, puede representar un riesgo considerable. Los cibercriminales pueden aprovecharse las redes públicas para llevar a cabo ataque..

Privacidad: Al solicitar datos personales para el acceso, pueden surgir preocupaciones en cuanto a la privacidad. Además, si no se manejan de forma adecuada. Podrían ser victima de una brecha de seguridad del sistema.

Experiencia del usuario: Algunos usuarios pueden encontrar molesto tener que pasar por un portal cautivo antes de poder acceder a la red.

4.7 Prospectiva

Mi vista es con el tiempo poder añadir como que se haga en un único sistema el cual hará todo, la mejora y optimización de la programación que dirige todo , el añadido de otros sistemas como de correo, y filtrado de datos y luego en una base de datos tener una visualización de lo que visitan los usuarios entre otras cosas.

4.8Presupuesto

Proyecto: Desarrollo de Portal Cautivo con el diseño de administración, y generación de token y administración de red

Descripción del Servicio	Cantidad /horas	Precio Unitario (€)	Total (€)
Diseño de interfaz de usuario	30	20	600
Pruebas y depuración	20	45	900
Infraestructura informatica		613	613
Implementación y puesta en marcha	10	50	500
Capacitación del personal	1	35	35
Soporte post-implementación	20	30	600
Total	-	-	3248

Dispositivos informáticos:



Qotom-Mini PC 5 x I225-V B3 2,5G Lan N4000 J4125

Frecuencia de CPU: Intel 2GHz

RAM: 16GB DDR4

Connectividad: 5 puertos Ethernet

Precio: **235,00€**

4.Aplicabilidad

Enlace:

<https://es.aliexpress.com/item/1005003926574260.html>

Leotec Mini Pc Intel Pentium N100/8GB/128GB SSD



Frecuencia de CPU: Intel Pentium N 2GHz

RAM: 8GB DDR4

Connectividad: 2 puertos Ethernet

Precio: 249,06€

Enlace: <https://www.pccomponentes.com/leotec-mini-pc-intel-pentium-n100-8gb-128gb-ssd>

ROUTER /SWITCH



**Mikrotik RB4011IGS+RM Router Ethernet 10 Puertos
RJ45 Gigabit PoE + 1 Puerto SFP+ 10G**

RAM: 1GB DDR3

Connectividad: 10 Ethernet lan

Precio: **213,04€**

Enlace: <https://www.pccomponentes.com/mikrotik-rb4011igs-rm-router-ethernet-10-puertos-rj45-gigabit-poe-1-puerto-sfp-10g>

Justificación del Presupuesto

Precio por hora: Hemos calculado el precio por hora teniendo en cuenta la experiencia y la especialización necesarias para cada tarea, así como los costo y de mantenimiento del equipo.

Diseño de interfaz de usuario: Creamos una interfaz intuitiva y atractiva para mejorar la experiencia del usuario.

Pruebas y depuración: Nos aseguramos de la calidad del software mediante pruebas exhaustivas y corrección de errores.

Implementación y puesta en marcha: Instalamos y configuramos el sistema en el entorno del cliente.

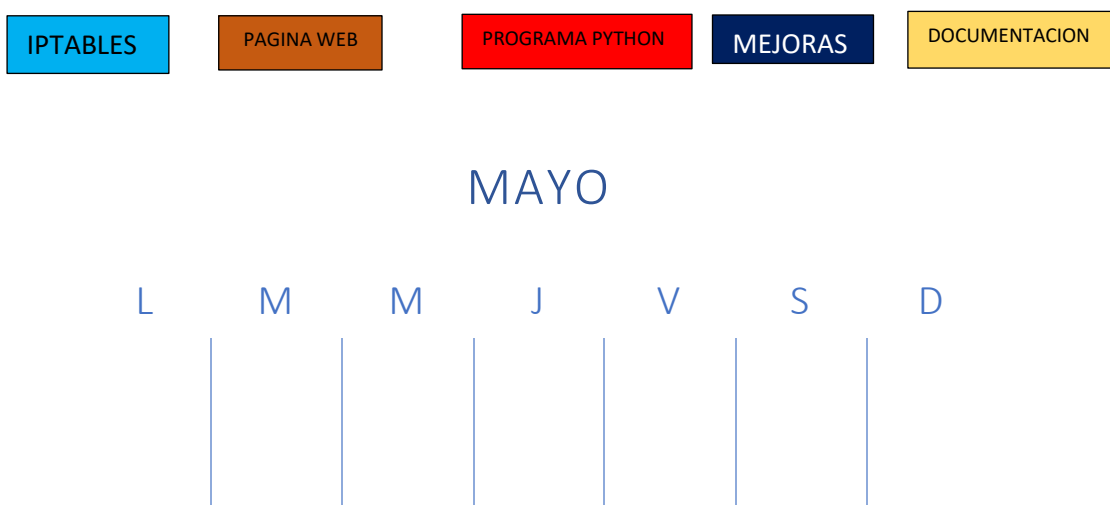
Capacitación del personal: Ofrecemos formación sobre el uso y mantenimiento del sistema para el personal.

Soporte post-implementación: Proporcionamos asistencia continua para resolver problemas y ofrecer actualizaciones.

El tiempo estimado para completar el proyecto es de 4 semanas a partir del inicio.

Es necesario que el cliente proporcione la información necesaria en un plazo de 2 semanas para evitar retrasos en el proyecto.

4.9 Previsio temporitzacio de tasques



4.Aplicabilidad

22	23	24	25	26	27	28
29	30					

ABRIL

L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

4.9 Conclusion

El objetivo del desarrollo de un portal cautivo es dar un acceso seguro y controlado a Internet en lugares públicos. El sistema utiliza tecnologías como iptables, PHP y Python para autenticar usuarios, controlar el acceso, implementar políticas de seguridad, y monitorear y registrar la actividad del usuario.

la seguridad es un uno de los principales desafíos, experiencia del usuario ofrece un sistema fácilmente entendible además de su simplicidad.

El proyecto tiene un gran potencial para ser útil para los administradores de red, a los proveedores de Internet que podrían implementarlo en lugares grandes.

5.Bibliografia

Ideas

<https://es.stackoverflow.com/questions/167508/portal-cautivo-mediante-iptables>

<https://jonathansandovalf.medium.com/cómo-hacer-un-portal-cautivo-de-la-muerte-c9b6f4d83437>

Programacion

<https://www.w3schools.com>

<https://ellibrodepython.com>

Funcionamiento portal cautivo Windows y Android

<https://unix.stackexchange.com/questions/386242/implementing-a-captive-portal-using-apache/386243#386243>

<https://serverfault.com/questions/459477/how-to-remove-a-mark-set-by-iptables>

<https://www.php.net/manual/es/function.ssh2-connect.php>

<https://stackoverflow.com/questions/75073724/call-to-undefined-function-ssh2-connect-after-php-8-1-14-update>

Velocidad

<https://serverfault.com/questions/833862/how-to-limit-speed-for-every-device-per-mac-address-in-the-gateway-via-linux-com>

6.Anexos

Video: <https://youtu.be/HILLwR2XqCs>

Canva: https://www.canva.com/design/DAGCBXvt9so/csofGGn6mhnMhI-H-UjCew/edit?utm_content=DAGCBXvt9so&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Github: <https://github.com/Lilarkzuli/PORTAL-CAUTIVO-SEMI-RARO>

Parar servidor:

```
Sudo systemctl stop portal.service
```

Arrancar servidor:

```
Sudo systemctl start portal.service
```

-Gracias a mi mama y mi perrita Kika, por aguantarme todo el rato mientras sufría con esto.

Gracias al que me enseñó Python, gracias al que me enseñó apache y gracias al que me enseñó iptables.

